PB-237 692

DEVELOPMENT OF MODELING TECHNIQUE FOR PHOTOCHEMICAL AIR POLLUTION

SYSTEMS, SCIENCE AND SOFTWARE

PREPARED FOR
NATIONAL ENVIRONMENTAL RESEARCH CENTER

FEBRUARY 1974

DISTRIBUTED BY:

NTIS
National Technical Information Service
U. S. DEPARTMENT OF COMMERCE

TECHNICAL REPORT DATA
(Please read Instructions on the reverse before completing)

1. REPORT NO. EPA-650/4-74-003	2. PB 237 692
4. TITLE AND SUBTITLE: Development of Modeling Technique for Photochemical Air Pollution	5. REPORT DATE: February 1974 6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S): L. H. Teuscher and L. E. Hauser	8. PERFORMING ORGANIZATION REPORT NO.
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Systems, Science and Software P. O. Box 1620 La Jolla, Ca. 92037	10. PROGRAM ELEMENT NO. 1A1009 11. CONTRACT/GRANT NO. 68-02-0272
12. SPONSORING AGENCY NAME AND ADDRESS: Meteorology Laboratory, EPA National Environmental Research Center Research Triangle Park, N. C. 27711	13. TYPE OF REPORT AND PERIOD COVERED: Final Report 14. SPONSORING AGENCY CODE
15. SUPPLEMENTARY NOTES	

16. ABSTRACT

A new particle-in-cell code has been developed and applied to the simulation of photochemical air pollution in the Los Angeles basin. The method is a Lagrangian one in which a parcel of air is followed and the chemistry takes place within it, in contrast to an earlier method in which the photochemical kinetics is considered in a three-dimensional space of fixed Eulerian cells. Although it can be shown that inherent errors are associated with the Eulerian scheme of computation it has not previously been clear how important these errors would be in actual simulations, when compared to the results of Lagrangian chemistry. In the present study the two methods were compared with actual measurements of photochemical air pollution for a selected day in the Los Angeles basin. The results obtained indicated that in the present rough state-of-the-art as regards modeling the chemical kinetics there is little to choose between the two methods. Some calculations were also made to test the sensitivity of the results to the input data. These emphasized the non-linearities of the problem and the dangers in extrapolating air quality levels simply from changes in emissions. The report documents the new Lagrangian code and provides a users guide to its operation.

17. KEY WORDS AND DOCUMENT ANALYSIS

a. DESCRIPTORS	b. IDENTIFIERS/OPEN ENDED TERMS	c. COSATI Field/Group
Air Pollution Mathematical Modeling	PRICES SUBJECT TO CHANGE	

18. DISTRIBUTION STATEMENT	19. SECURITY CLASS *(This Report)* Unclassified	21. NO. OF PAGES 99
	20. SECURITY CLASS *(This page)* Unclassified	22. PRICE 4.75-2.25

Form 2220-1 (9-73)

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
U S Department of Commerce
Springfield VA 22151

EPA-650/4-74-003

DEVELOPMENT OF MODELING TECHNIQUE FOR PHOTOCHEMICAL AIR POLLUTION

by

L. H. Teuscher and L. E. Hauser

Systems, Science and Software
P. O. Box 1620
La Jolla, California 92037

Contract No. 68-02-0272
Program Element No. 1A1009

EPA Project Officer: Kenneth L. Calder

Meteorology Laboratory
National Environmental Research Center
Research Triangle Park, North Carolina 27711

Prepared for

OFFICE OF RESEARCH AND DEVELOPMENT
U.S. ENVIRONMENTAL PROTECTION AGENCY
WASHINGTON, D.C. 20460

February 1974

This report has been reviewed by the Environmental Protection Agency and approved for publication. Approval does not signify that the contents necessarily reflect the views and policies of the Agency, nor does mention of trade names or commercial products constitute endorsement or recommendation for use.

TABLE OF CONTENTS

Page

PART I – THEORY AND TECHNICAL RESULTS

1. INTRODUCTION 1

1.1 Errors in Eulerian Chemistry 2

1.2 Initial Evaluation of Errors for Observed Concentrations 4

2. DESCRIPTION OF THE LAGRANGIAN METHOD 7

2.1 The Diffusion Equation 7

2.2 Particle Advection 8

2.3 Eulerian Cell Concentration 12

2.4 Sources 13

2.5 Chemical Reactions 16

2.6 General Code Operation 16

3. TESTING OF THE NEXUS/L CODE 18

4. PHOTOCHEMICAL MECHANISM AND SIMULATION 21

5. SUMMARY AND CONCLUSIONS 39

References 40

PART II – OPERATIONS MANUAL

1. INTRODUCTION 41

2. DISCUSSION OF THE NEXUS/L CODE 42

2.1 Grid Labeling Conventions 42

2.2 The Nexus/L Particles 42

2.3 Flow Logic of the Nexus/L Code 44

2.4 Glossary of Key Quantities 48

TABLE OF CONTENTS, contd.

Page

2.5 How Problems are Generated 49

2.5.1 Namelist "START" 49

2.5.2 Namelist "SPECS" 49

2.5.3 Namelist "GEN" 52

3. CODE LISTING 54

4. A SAMPLE TEST CALCULATION 85

4.1 Code Changes 85

4.2 Input to Generate Test Calculation 86

4.3 Cycle 1 Output 88

Reference 88

SSS-R-74-1756

PART I

THEORY AND TECHNICAL RESULTS

P.O. BOX 1620, LA JOLLA, CALIFORNIA 92037, TELEPHONE (714) 453-0060

1. INTRODUCTION

During the past years, the Environmental Protection Agency (EPA) and predecessor organizations have funded the development of photochemical simulation models which are to be used in the evaluation of present photochemical air pollution problems, the prediction of potential control strategies and the impact of new industrial plant siting on an urban region.

Two general methods of approach have evolved for photochemical simulation. The first model, typified by the models developed by Eschenroeder, et.al. (1971) and System Development Corporation (1970) is a Lagrangian method where a parcel of air is followed and chemistry takes place inside the packet of air. The other approach that has been considered is an Eulerian approach where the three-dimensional space is divided into subvolumes or cells and during each time increment pollutants are allowed to move from cell to cell by wind advection and diffusion. Roth, et.al. (1971) have developed a model along these lines. Systems, Science and Software (1971) developed a composite model using the PICK method (Particle-In-Cell with K-theory diffusion) which has Lagrangian characteristics but does chemistry in fixed Eulerian cells.

The objective of the research effort described in this report was to extend the model developed by Systems, Science and Software (S^3) so that the photochemical kinetics can be considered in a Lagrangian framework and comparisons between

results using Eulerian and Lagrangian chemistry can be made to determine in actual practice if significant differences between the methods occur. It can be shown that some inherent errors do exist in Eulerian chemistry; however, it is not clear how important quantitatively these errors will be when compared to the results of the Lagrangian chemistry in actual simulations.

1.1 ERRORS IN EULERIAN CHEMISTRY

This illustration of the basis for the inherent errors in Eulerian chemistry was originally presented in the S^3 Final Report in partial fulfillment of EPA Contract 68-02-0006.

Consider a one-dimensional problem as illustrated in Figure 1.

CELL	1	2	3	4	5	6
	0.0	1.0	1.0	1.0	0.0	0.0

Figure 1. Initial distribution of primary pollutant A.

Initially, cells 2-4 each have only primary pollutant, A, with cell concentration equal to 1.0. The effects of advection for a wind blowing four-tenths of a cell in time step Δt are simulated first, resulting in the distribution shown in Figure 2. If a quadratic chemical mechanism ($A + A \rightarrow B$) is hypothesized to illustrate the non-linearities in chemical kinetics rate equations, the rate equations can be written as

CELL	1	2	3	4	5	6
	0.0	0.6	1.0	1.0	0.4	0.0

Figure 2. Distribution of primary pollutant A after advection.

$$\frac{d[A]}{dt} = -k[A]^2 \qquad \frac{d[B]}{dt} = +k[A]^2 \quad .$$

Furthermore, if k is small, then $k[A]^2 \Delta t << [A]$ and the effects of the chemical reaction on the concentration of A can be ignored in order to focus on the secondary pollutant production. With $[A]$ approximately constant during the reactions the equation for production of $[B]$ can be integrated

$$[B] = k[A]^2 \Delta t \quad .$$

That is, the production of the secondary pollutant is proportional to the square of the concentration of the primary pollutant. If, as an Eulerian technique would require, the cell concentrations of A are used in this equation, the distribution of B would be as shown in Figure 3.

CELL	1	2	3	4	5	6
	0.0	0.36 $k\Delta t$	1.0 $k\Delta t$	1.0 $k\Delta t$	0.16 $k\Delta t$	0.0

Figure 3. Distribution of secondary pollutant, B, calculated from cell concentrations given in Figure 2.

On the other hand, the concentration of A inside the puff is 1.0 and outside 0.0. That is, in a Lagrangian frame the production of B would be as shown in Figure 4.

CELL	1	2	3	4	5	6
	0.0	0.6 kΔt	1.0 kΔt	1.0 kΔt	0.4 kΔt	0.0

Figure 4. Distribution of secondary pollutant, B, calculated from Lagrangian concentration.

This is the physically correct picture. The methodology used in chemical simulation with Eulerian cell concentrations introduced errors of 40% in cell 2 and 60% in cell 5 when compared to the correct Figure 4.

1.2 INITIAL EVALUATION OF ERRORS FOR OBSERVED CONCENTRATIONS

In this section, observed concentrations at the Downtown Los Angeles and Pasadena stations of the Los Angeles County Air Pollution Control District (APCD) are used to illustrate and evaluate the Eulerian concentration errors. At 7:00 a.m., on September 30, 1969, the following concentrations were observed:

	Downtown Los Angeles	Pasadena
NO	51 pphm	19 pphm
NO_2	1 pphm	9 pphm
HC	9 ppm	5 ppm
O_3	3 pphm	1 pphm

These sets of concentrations can be used as initial conditions for a chemical reaction simulation. In this illustration, the simulation uses the CHEM computer code as described in the previously referenced S[3] Final Report. The results after 5 and 10 minutes are shown in the first column of Table I.

TABLE I

TIME	POLLUTANT	LOS ANGELES	PASADENA	MIXED CELL	
				LAGRANGIAN	EULERIAN
0	NO	51.0	19.0	19.0	19.0
	NO_2	1.0	9.0	9.0	9.0
	HC	9.0	5.0	5.0	5.0
	O_3	3.0	1.0	1.0	1.0
5	NO	46.0	16.0	28.0	27.0
	NO_2	5.8	11.0	9.2	10.0
	HC	9.0	5.0	6.6	6.6
	O_3	0.02	0.1	0.08	0.06
10	NO	43.0	14.0	37.0	37.0
	NO_2	8.9	14.0	9.8	7.8
	HC	9.0	5.0	8.2	8.2
	O_3	0.04	0.2	0.07	0.03

The Lagrangian and Eulerian results listed under "Mixed Cell" correspond to a position intermediate between Downtown Los Angeles and Pasadena. Initially, the air at this position corresponds to that over Pasadena. It has been assumed for this illustration that "Downtown"-type air replaces four-tenths of the "Pasadena" air over this position each five

minutes. Corresponding to Figure 1, the position being examined is similar to cells 2 or 5. The Lagrangian results are just the proper proportions of the two types of air. The Eulerian results show the effects of performing the chemical simulation with "mixed" cell concentrations – an error of approximately 10% in O_3 and 20% in NO_2 in only two cycles (10 minutes). After a few hours errors inherent in the Eulerian simulation may be compounded and the results may be meaningless or, on the other hand, the errors may tend to average out.

This illustration is, like the first illustration, a gross over-simplification. These inherent errors dominate only where large gradients are present. The objective of the present research is to examine the potential differences by completing comparison calculations of the Los Angeles basin using both the Eulerian and Lagrangian chemistry, and then evaluating the two techniques if significant differences occur.

2. DESCRIPTION OF THE LAGRANGIAN METHOD

2.1 THE DIFFUSION EQUATION

In this report a Lagrangian approach to the solution of the "K-theory" equation describing the advection and dispersion of pollutants in the atmosphere is presented. In a previous study, Systems, Science and Software (1971) describes the PICK method where each particle represents an amount of mass which is transported by winds and dispersion is accounted for by a "turbulent flux velocity." The PICK method has been applied to two- and three-dimensional air pollution problems with good results. Artificial dispersion usually found in Eulerian codes is greatly reduced and good definition of distributions can be maintained.

In the Lagrangian approach, it is necessary that the particles do not represent mass but represent concentration of pollutant so that chemistry can be accomplished within each particle as it moves along. The basic ideas of particle representation used by S^3 (1971) are utilized in the development of the new computer simulation model.

The basic Eulerian equation which is to be solved is a set of coupled K-theory equations for each chemical specie as shown in Eq. (1).

$$\frac{\partial C_i}{\partial t} + u\frac{\partial C_i}{\partial x} + v\frac{\partial C_i}{\partial y} + w\frac{\partial C_i}{\partial z} = \frac{\partial}{\partial x}\left(K_x \frac{\partial C_i}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y \frac{\partial C_i}{\partial y}\right)$$

$$+ \frac{\partial}{\partial z}\left(K_z \frac{\partial C_i}{\partial z}\right) + \left(\frac{\partial C_i}{\partial t}\right)_{chem} + S_i \qquad (1)$$

where

C_i = concentration of ith specie

x,y,z = Cartesian coordinates

u,v,w = wind velocities

K_x, K_y, K_z = diffusivities

S_i = source rate of ith specie

$\left(\frac{\partial C_i}{\partial t}\right)_{chem}$ = rate of change of ith specie due to chemical reactions

Equation (1) is not completely general, since the diffusivity should be considered as a tensor and the x,y,z axes are not necessarily principle axes as would be inferred from Eq. (1). However, the current knowledge of diffusivity variation in the atmosphere does not allow a more detailed description than implied by Eq. (1) so the developed model does not consider a more general formulation.

2.2 PARTICLE ADVECTION

In the Lagrangian PICK code each particle represents a specific concentration; that is, associated with each particle is a concentration of pollutant. Therefore, the concentration in an (Eulerian) cell can be determined by appropriately averaging over the total numbers of particles in the cell. The averaging method used is discussed in Section 2.3. If more than one pollutant is being considered, such as in photochemical

simulations, then concentrations of several pollutant species can be associated with the same particle. For example, a single particle could be associated with the concentrations of HC, NO, NO_2, HNO_2 and O_3. Chemical reactions would be carried out for each particle as it travels through the grid. Thus, the description of the change of pollutant concentrations due to the chemical reactions and advection will be described in a purely Lagrangian manner. However, for the purpose of (1) editing the results, (2) introducing pollutant sources into the problem, (3) calculating diffusion fluxes, and (4) inputing a wind field, an Eulerian grid system is still used. This grid system is identical to that used in NEXUS and the method of computing particle velocity by volume averaging is identical to that used for calculating particle velocity in NEXUS. Each cell has a velocity vector located at the cell center. The particles in the cells are then moved for each time step with a velocity obtained by linear interpolation according to the position of the particle between the centers of adjacent cells. This is illustrated for two dimensions in Figure 5. Using $\vec{v}^p$ to denote the total particle velocity vector (u,v)

$$\vec{v}^p(x,y) = \left\{(y-y_j)\left[\frac{\vec{v}_{i+1,j+1}(x-x_i) + \vec{v}_{i,j+1}(x_{i+1}-x)}{\Delta x}\right] + (y_{j+1}-y)\left[\frac{\vec{v}_{i+1,j}(x-x_i) + \vec{v}_{i,j}(x_{i+1}-x)}{\Delta x}\right]\right\}\Big/\Delta y \qquad (2)$$

The shaded rectangular area in Figure 5 is cell-sized and centered at the particle position. Rearranging the terms in Eq. (2) gives:

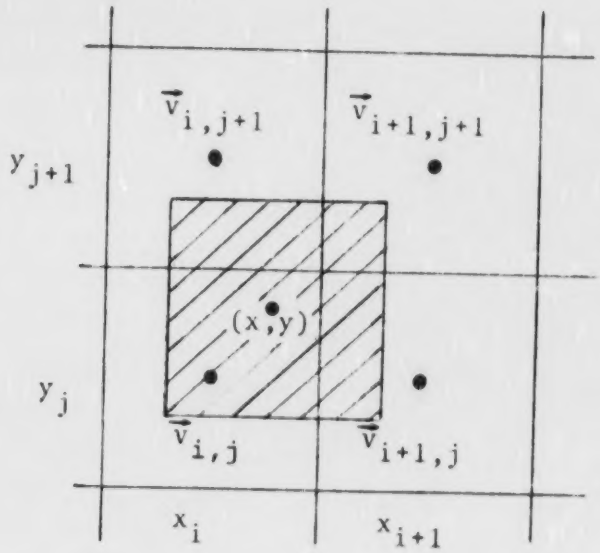


Figure 5. Area weighting interpolation for total velocity.

$$\vec{v}^p(x,y) = \frac{1}{\Delta x \Delta y}\,(x-x_i)(y-y_j)\vec{v}_{i+1,j+1} + (x_{i+1}-x)(y-y_j)\vec{v}_{i,j+1} + (x-x_i)(y_{j+1}-y)\vec{v}_{i+1,j} + (x_{i+1}-x)(y_{j+1}-y)\vec{v}_{i,j} \quad . \tag{3}$$

The product $(x-x_i)(y-y_j)$ multiplying $\vec{v}_{i+1,j+1}$ is the shaded area overlapping cell $(i+1,j+1)$ in Figure 5. Similarly, the other products give the areas of the overlap with the other cells. To simplify notation, the commas used to separate the subscripts will be suppressed for the remainder of this report.

In three dimensions, the analogous volume overlap is used. Therefore, in three dimensions, the velocity for a particle would be determined by the volume fraction in each of the surrounding cells. This would be computed as:

$$\vec{v}^{p} = \sum_{ijk} \vec{v}_{ijk} \, f^{p}_{ijk}$$

where

$\vec{v}^{p}$ = particle velocity

$\vec{v}_{ijk}$ = velocity in cell ijk

f^{p}_{ijk} = the fraction of the pth particle volume in the ijk cell

The sum is over the cells (ijk) where f^{p}_{ijk} is non-zero. This will normally be eight cells. (For example, four in a plane as shown in Figure 5 and four in the level above or below).

In the PICK method, the calculation to advance the particle configuration in time proceeds in steps or cycles, each of which calculates the desired quantities for time $t+\Delta t$ in terms of those at time t (an "explicit" time advancement procedure)

$$x(t+\Delta t) = x(t) + u\Delta t$$

$$y(t+\Delta t) = y(t) + v\Delta t \quad . \tag{4}$$

The velocities are the total velocities determined for the beginning of the time interval and interpolated to initial particle positions. These are held constant throughout any elementary time interval. It is evident that the magnitude

of the time interval Δt must be restricted, as otherwise a particle could pass through many cells in a cycle and well out of the region for which its velocity was interpolated. This could result in large inaccuracies and instabilities in the solution. Limiting the time step so that no particle moves more than four-tenths of a cell within any one cycle has been used as an empirical rule to avoid this problem.

2.3 EULERIAN CELL CONCENTRATION

The pollutant concentration in each cell is also computed by volume averaging of the particle concentrations. Since each particle carries concentration, it is necessary to have particles everywhere in the domain of interest. The method of computing the Eulerian cell concentrations is to associate with each particle a volume equal to a cell volume. It is not necessary to choose the particle volume to be equal to the cell volume; any choice would be valid. However, if the particle volume were equal to several cell volumes, the pollutant distributions would be necessarily spread out and "smeared." The other choice would be to choose a particle volume smaller than a cell volume. This would increase the number of particles necessary to carry out a calculation, since sufficient particles are required to cover all of the Eulerian grid. It would also increase the computational complexity. Since the fixed grid is governing all other aspects of the problem, such as velocity input and source input, it seems reasonable to assume a particle volume equal to a cell volume.

The Eulerian cell concentration is calculated by computing a volume average from Eq. (5):

$$C_{ijk} = \frac{\sum_{p=1}^{NP} c^p f^p_{ijk}}{\sum_{p=1}^{NP} f^p_{ijk}} \qquad (5)$$

where

C_{ijk} = concentration in cell ijk

c^p = particle concentration

NP = total number of particles in the problem

f^p_{ijk} = the fraction of the pth particle volume in the ijk cell

2.4 SOURCES

The method of attributing sources to a particle is almost the reverse procedure. All sources are treated as volume sources and are expressed in units of concentration per unit time (for example, ppm/min). As with any grid method, the smallest resolvable scale is one cell. Thus, treatment of sources as volume sources is not a serious deficiency. The method of computing the source rate for each particle is to take the fraction of the particle volume in the cell times the source rate of that cell to determine the particle source rate. This procedure is followed for all cells within which a portion of the particle volume lies, resulting in the expression:

$$s^p = \sum_i \sum_j \sum_k s_{ijk} f^p_{ijk} \qquad (6)$$

where

S_{ijk} = source strength in ijk cell

f^{p}_{ijk} = fraction of pth particle volume in cell ijk

For inert pollutants, the change in concentration of a particle due to sources is simply computed by multiplying the particle source rate by the total time step of the computation cycle. For photochemical pollutants, the source is incorporated into the chemistry routine.

In this model, diffusion is also treated as a source or sink. After the concentration is obtained in each of the Eulerian cells, then the time rate of concentration change in each Eulerian cell is computed using the usual centered difference scheme. The resulting difference equation is:

$$D_{S_{ijk}} = \left[\frac{K_{i+\frac{1}{2}jk}\left(\frac{C_{i+1jk}-C_{ijk}}{\Delta x}\right) - K_{i-\frac{1}{2}jk}\left(\frac{C_{ijk}-C_{i-1jk}}{\Delta x}\right)}{\Delta x}\right] + \left[\frac{K_{ij+\frac{1}{2}k}\left(\frac{C_{ij+1k}-C_{ijk}}{\Delta y}\right) - K_{ij-\frac{1}{2}k}\left(\frac{C_{ijk}-C_{ij-1k}}{\Delta y}\right)}{\Delta y}\right] + \left[\frac{K_{ijk+\frac{1}{2}}\left(\frac{C_{ijk+1}-C_{ijk}}{\Delta z}\right) - K_{ijk-\frac{1}{2}}\left(\frac{C_{ijk}-C_{ijk-1}}{\Delta z}\right)}{\Delta z}\right] \quad (7)$$

where

$D_{S_{ijk}}$ = diffusion source in the ijk cell

C_{ijk} = concentration in the ijk cell

K_{ijk} = diffusivity at center of cell ijk (subscripts in Eq. (7) with ±½ added to the integer subscripts indicate boundary values; see Figure 6)

$\Delta x, \Delta y, \Delta z$ = cell dimensions

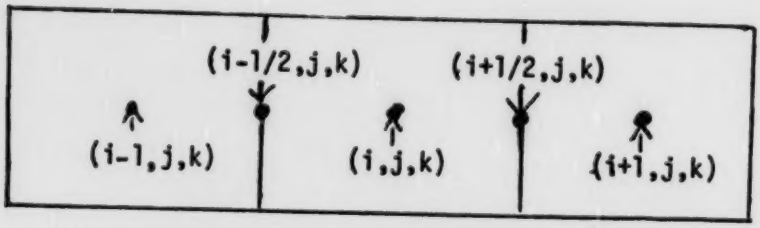


Figure 6. Location of centers of cells and interfaces in (i,j,k) notation.

The Lagrangian version of Eq. (1) is assumed to hold for each Lagrangian particle in the system, so that the equation solved for each particle is:

$$\frac{DC_i^p}{Dt} = \left.\frac{\partial C_i}{\partial t}\right)_{chem} + S^p_{sources} \quad . \tag{8}$$

where $\frac{D}{Dt}$ represents the time rate of change moving with a fluid particle (which is represented by the particles used in the numerical method. $S^p_{sources}$ represents the particle source rates, including diffusion, which are computed from the cell source rates using Eq. (6). Thus, the change in concentration of a particle is due to chemistry, diffusion and sources.

2.5 CHEMICAL REACTIONS

As in the NEXUS/P code developed by S^3 (1972), the chemistry module is such that it is very simple to implement another chemical reaction system. The basic input to the chemistry routine is: (1) the concentration of each pollutant species, and (2) the source rate of the pollutant due to sources and diffusion. The chemistry routine then solves the coupled non-linear equations:

$$\frac{\partial C_i}{\partial t} = F_i - R_i C_i + S_i$$

where

$F_i = f_i(C_j)$ - formation rate of C_i

$R_i C_i = r_i(C_j) C_i$ - removal rate of C_i

S_i = source rate of C_i

2.6 GENERAL CODE OPERATION

Figure 7 depicts the logical sequence used in each cycle of calculation. Because each particle contains the concentration of all species, the total number of particles required in the calculations are less than required in a NEXUS/P calculation. Consequently, some savings in the use of auxiliary storage is provided.

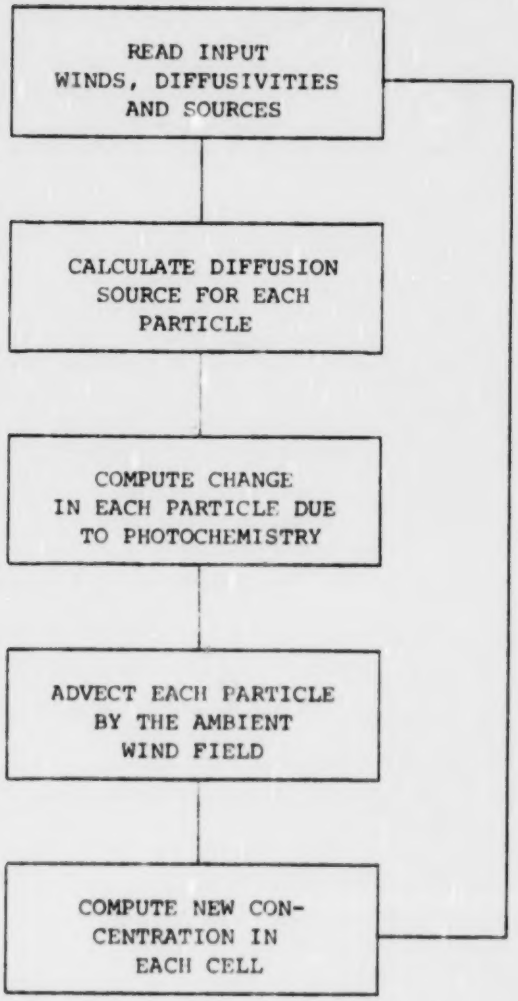


Figure 7. A typical cycle in the NEXUS/L code.

3. TESTING OF THE NEXUS/L CODE

Since each particle in the NEXUS/L code represents a concentration of pollutant, it is necessary to have sufficient particles distributed throughout the grid so that all space is occupied by at least one particle volume. Another difficulty with a Lagrangian code is the accuracy with which the distribution can be interpolated onto the Eulerian grid. If a sharply peaked distribution is being advected then the position of the gradients within a cell can cause the distribution to appear to be distorted. However, when the distribution returns to a position similar to the original, the Eulerian results will agree with the actual distribution.

In a previous report, S^3 (1971), describing the PICK method, a test problem used to study the accuracy of the code was the advection of a Gaussian distribution. The same test problem was considered using the NEXUS/L code. The distribution was defined initially by

$$C(x,y,0) = 71.1\{\exp -[(x-9)^2 + (y-9)^2/12.5]\} \quad .$$

The velocities were chosen as $\bar{v}_x$ = 10m/sec and $\bar{v}_y$ = 5 m/sec. Figure 8 shows the motion of the Gaussian distribution across the grid. There is no distortion of the distribution. The peak concentration in each cell remained at its initial concentration value. This particular test problem was run with 1, 2 and 4 particles in each cell. The results were independent

Reproduced from
best available copy.

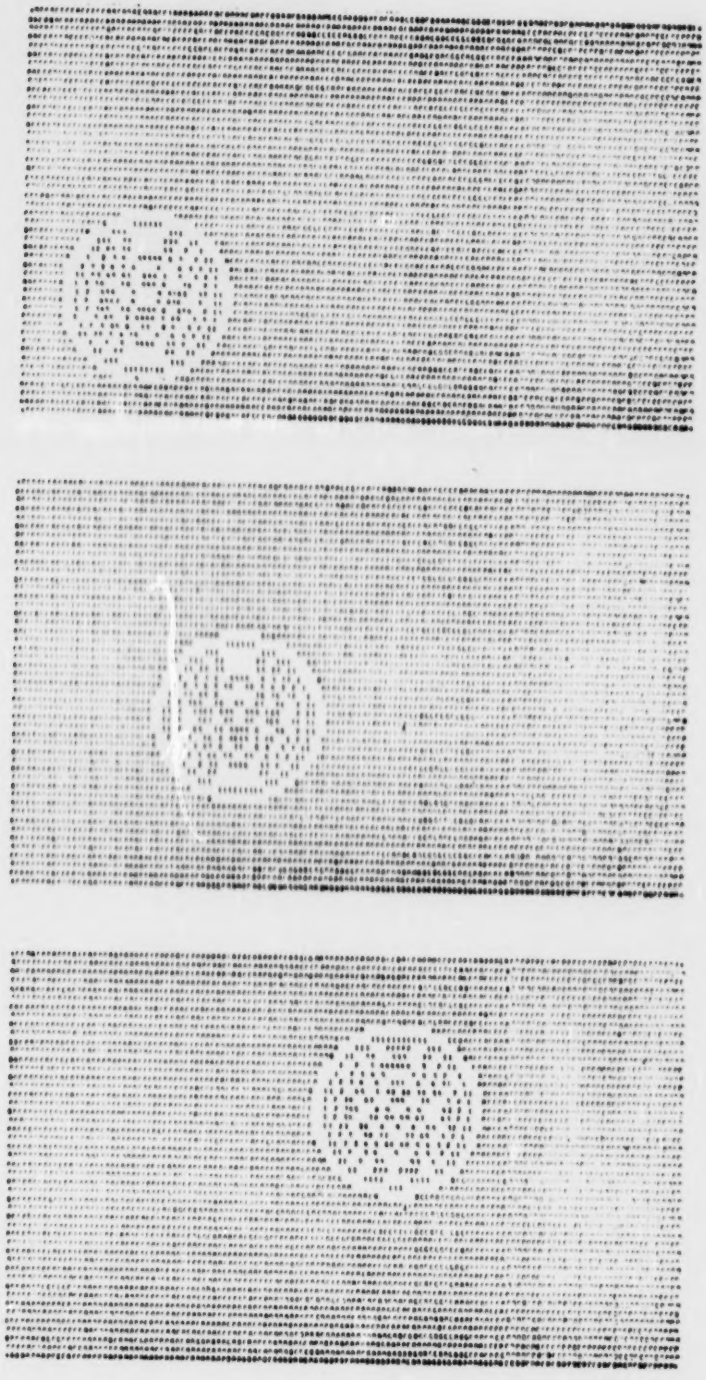


Figure 8. Advection of a Gaussian distribution v_x = 10 m/sec and v_y = 5 m/sec.

of the number of particles used. This test problem indicates that advection is properly treated using the modified particle method and the advantages attributed to the particle method applied in the modified approach.

4. PHOTOCHEMICAL MECHANISM AND SIMULATION

The photochemical mechanism used for the simulation results presented is a mechanism developed by Eschenroeder and Martinez (1972). Table II gives the mechanism and the rate coefficients used. This mechanism contains three branching factors, b_1,b_2,b_3 in Eqs. (3), (4) and (5), which account for the production of RO_2, the generalized oxidant radical, and one yield factor, y, in Eq. (6), which affects the amount of OH radical produced.

By carefully selecting these parameters, the resulting rate of change of pollutants can be made to agree with almost any desired result. The method of selecting these parameters was to examine the behavior of a "basin average" obtained by averaging the actual pollutant measurements from the APCD stations for the simulation day, September 30, 1969. An average source rate was determined for each pollutant and a series of chemical simulations was made varying the branching factors and yield factor to provide a reasonable fit to the average chemical behavior of the basin. The reason these "free" parameters exist is that the exact composition of the hydrocarbons present is not known and if it were, the chemistry required to include each hydrocarbon species would make the system too complex to be manageable. Thus, a "generalized lumped hydrocarbon" is used and consequently the "free" parameters appear and must be adjusted for any hydrocarbon mix. Eschenroeder and Martinez (1972) discuss this point at some length.

TABLE II

PHOTOCHEMICAL KINETICS SYSTEM

	Reaction	Rate Constants
1.	$(h\nu) + NO_2 \rightarrow NO + O$	$0.267\ min^{-1}$*
1a.	$O + (O_2) + M \rightarrow O_3 + (M)$	$2.64 \times 10^{6}\ min^{-1}$
2.	$NO + O_3 \rightarrow NO_2 + (O_2)$	$26.7\ ppm^{-1}\ min^{-1}$
3.	$O + HC \rightarrow (b_1)RO_2$	$2.81 \times 10^{-5}\ ppm^{-1}\ min^{-1}$
4.	$OH + HC \rightarrow (b_2)RO_2$	$1.5 \times 10^{4}\ ppm^{-1}\ min^{-1}$
5.	$O_3 + HC \rightarrow (b_3)RO_2$	$4.0 \times 10^{-3}\ ppm^{-1}\ min^{-1}$
6.	$RO_2 + NO \rightarrow NO_2 + (y)OH$	$1.0 \times 10^{5}\ ppm^{-1}\ min^{-1}$
7.	$RO_2 + NO_2 \rightarrow PAN$	$2.0 \times 10^{2}\ ppm^{-1}\ min^{-1}$
8.	$OH + NO \rightarrow HONO$	$1.5 \times 10^{3}\ ppm^{-1}\ min^{-1}$
9.	$OH + NO \rightarrow HNO_3$	$3.0 \times 10^{3}\ ppm^{-1}\ min^{-1}$
10.	$(h\nu) + HONO \rightarrow OH + NO$	$1.0 \times 10^{-3}\ min^{-1}$*
11.	$NO + NO_2 \xrightarrow{H_2O} 2HONO$	$1.0 \times 10^{-3}\ min^{-1}$
12.	$NO_2 + O_3 \rightarrow NO_3 + O_2$	$5.0 \times 10^{-2}\ ppm^{-1}\ min^{-1}$
13.	$NO_3 + NO_2 \rightarrow N_2O_5$	$4.5 \times 10^{3}\ ppm^{-1}\ min^{-1}$
14.	$N_2O_5 \rightarrow NO_3 + NO_2$	$1.4 \times 10^{1}\ ppm^{-1}\ min^{-1}$
15.	$N_2O_5 + (H_2O) \rightarrow 2HNO_3$	$6.05 \times 10^{1}\ min^{-1}$

where $b_1 = 10$, $b_2 = 10$, $b_3 = 0.1$, and $y = 0.1$.

*Corresponding to noon value of solar insolation.

Photochemical smog in the Los Angeles basin on September 30, 1969, was simulated using NEXUS/P. This calculation was a repeat of the calculation reported by S[3] (1971) with a revised chemical mechanism. The chemical mechanism used was the mechanism by Eschenroeder and Martinez, shown in Table II, with five calculated species (NO, NO_2, HC, O_3 and HNO_2) and three species in pseudo-equilibrium (O, RO_2 and OH). The source inventory used was reported by Roberts, et.al. (1971). The required meteorological wind data were obtained from Dickson and Start (1971) and Roth (1971) provided inversion height data. The 16-hour real time simulation required 1.5 hours on a UNIVAC 1108. Approximately one-half hour was spent in chemical simulation and one-half hour in data manipulation.

The NEXUS/L code was used to perform a comparison to the updated NEXUS/P calculation. The same winds, diffusivities and sources were used for this photochemical simulation. A comparison of the results provided by the two codes is shown in Figures 9 through 18. The available APCD data are also presented.

An examination of the basin averaged results, the average of results at all APCD stations, shown in Figure 9, yields some general conclusions regarding the differences exhibited between the two calculations. The NO_2 levels in the NEXUS/P calculations show a general trend of being larger than in the NEXUS/L calculation. The peak value is 15% higher. The NO_2 levels remain above the 10 pphm level throughout the afternoon in the NEXUS/P results, whereas the NO_2 disappears between 1:00 and 2:00 p.m. in the NEXUS/L results. The disparity between the NO_2 results from each calculation causes the oxidant levels, represented by O_3 in the chemical model, to also be different in the two calculations. The basin average results indicate peak values 30% higher for the NEXUS/L calculation and the peak values occur 1-1½ hours later in the day.

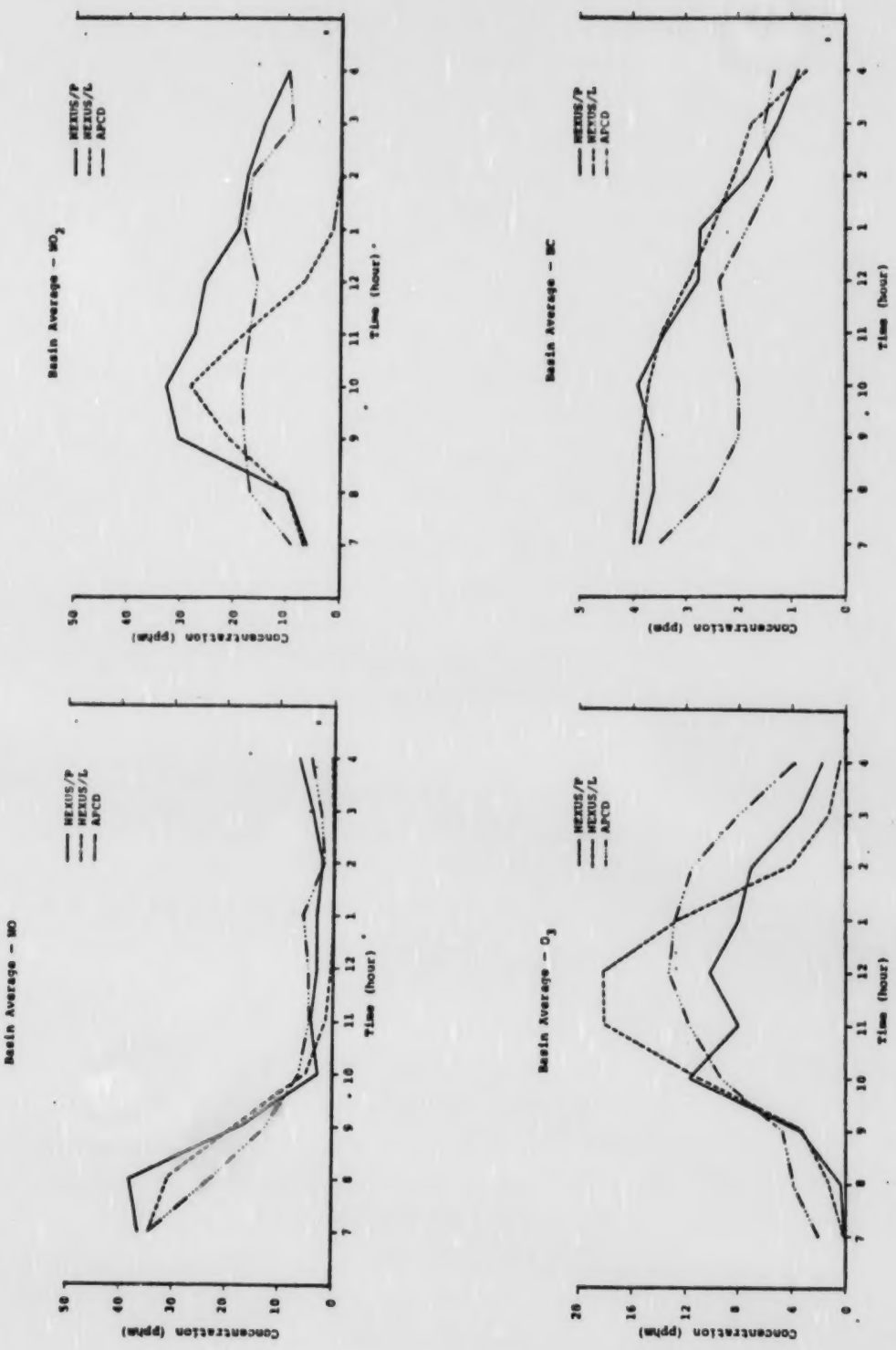


Figure 9. Basin average pollutant concentrations.

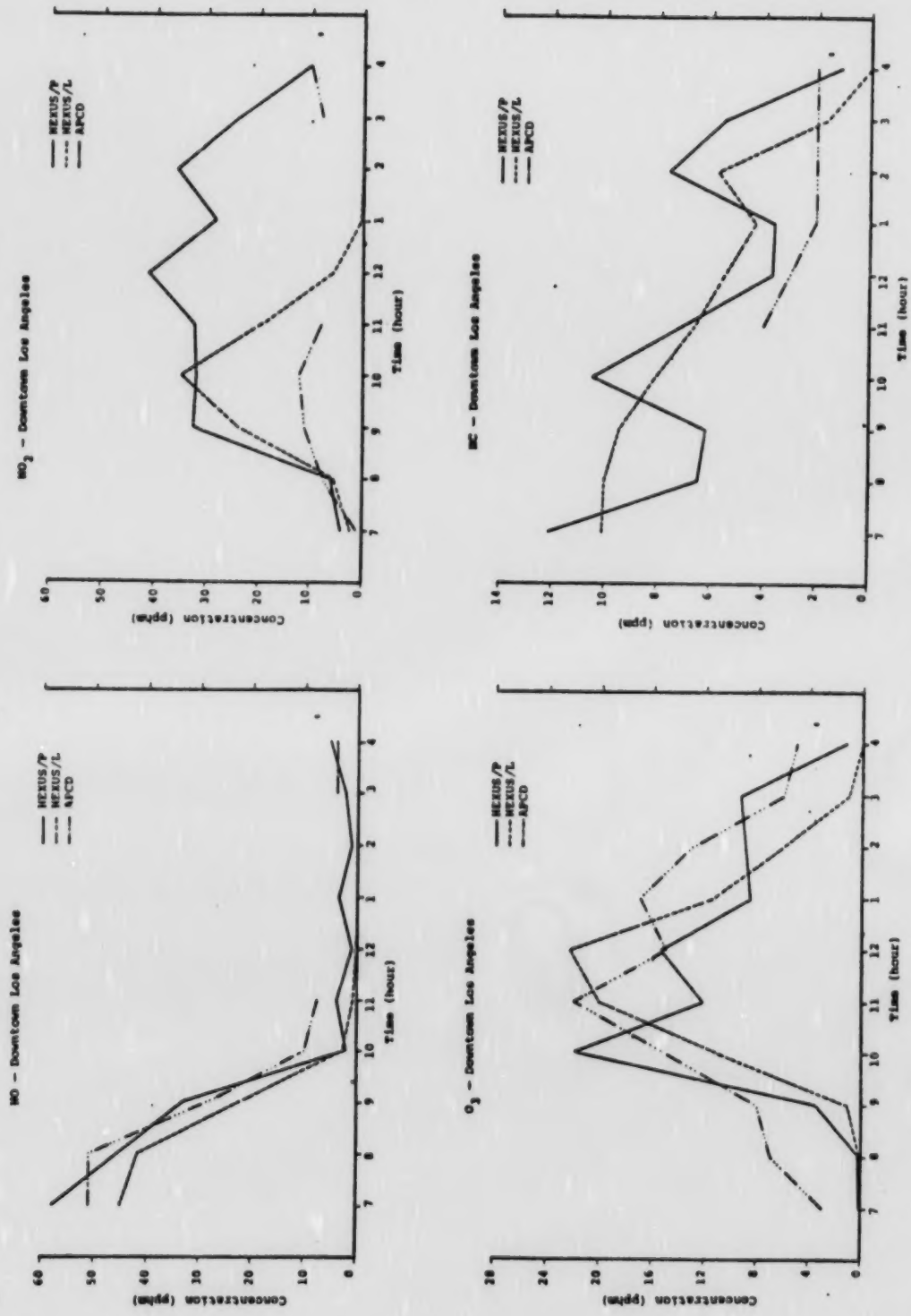


Figure 10. Downtown Los Angeles pollutant concentrations.

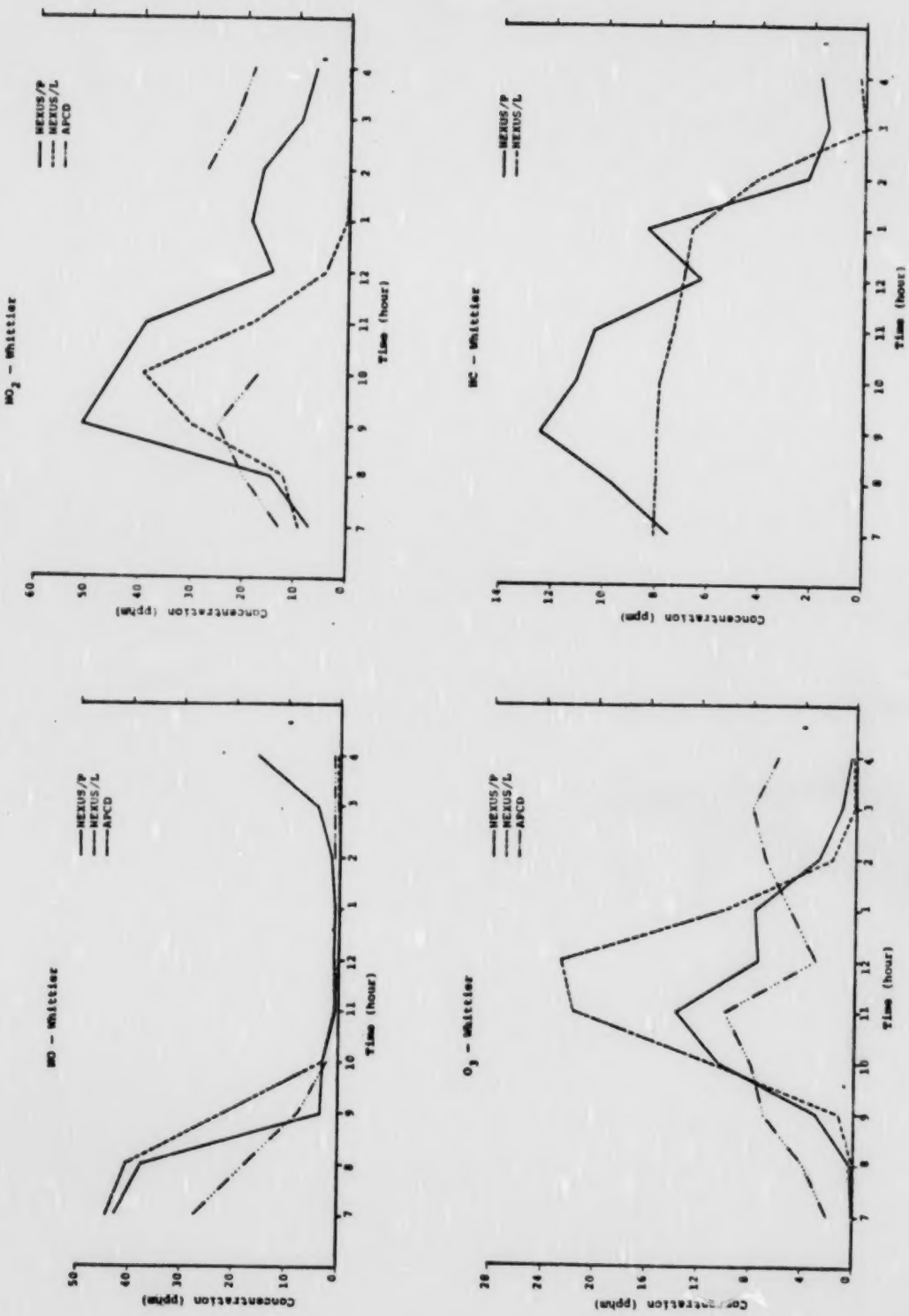


Figure 11. Whittier pollutant concentrations.

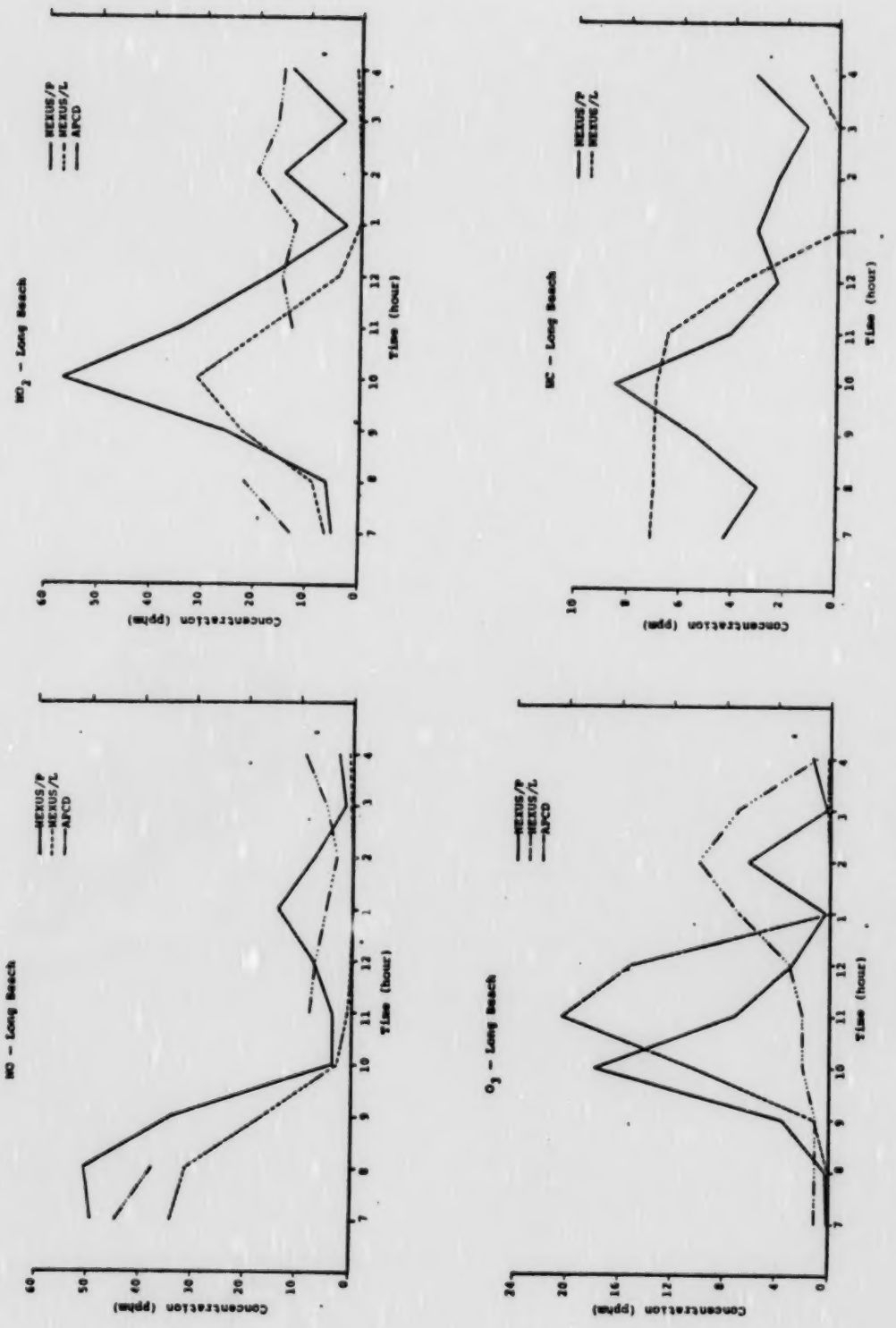


Figure 12. Long Beach pollutant concentrations.

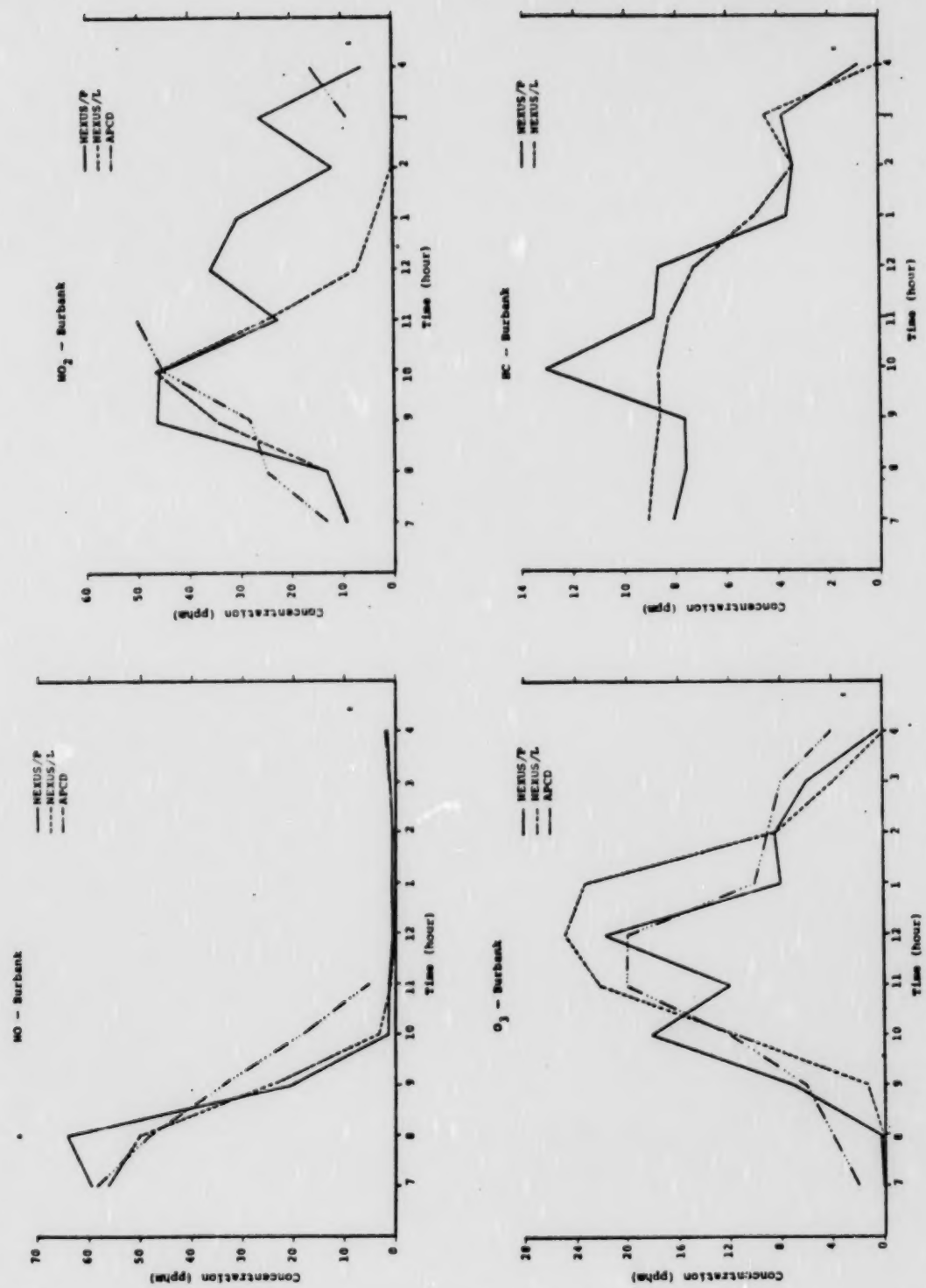


Figure 13. Burbank pollutant concentrations.

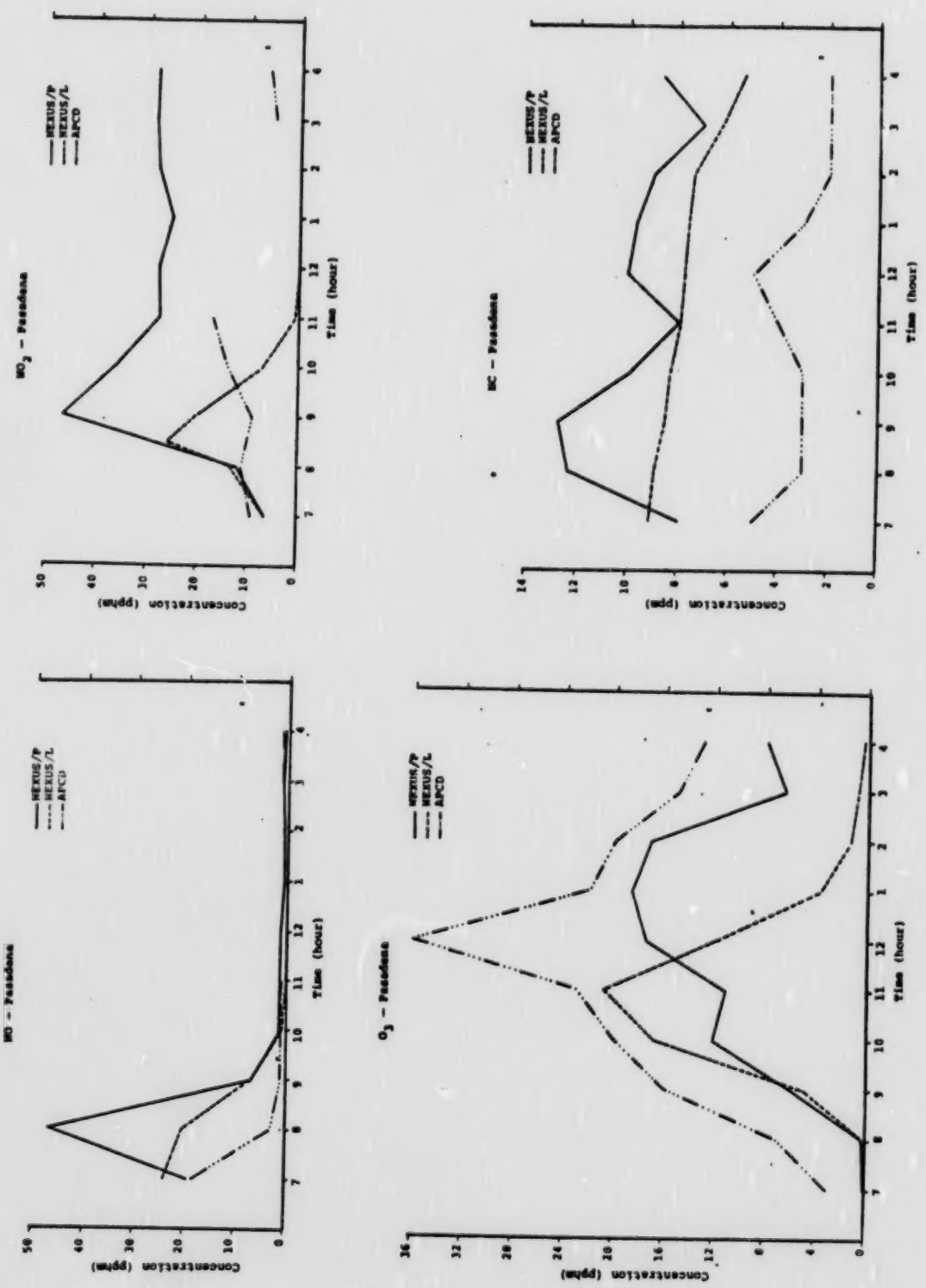


Figure 14. Pasadena pollutant concentrations.

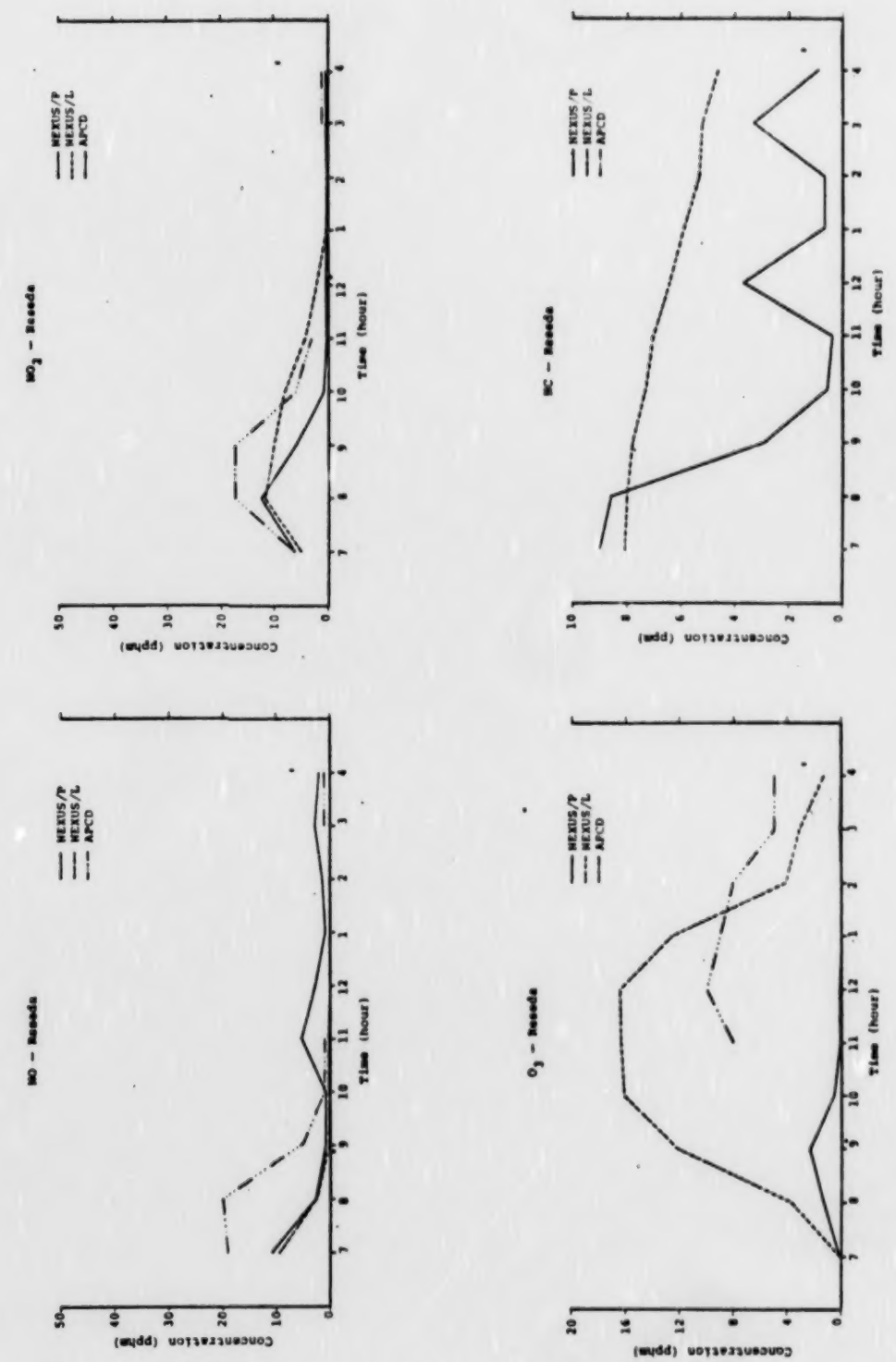


Figure 15. Reseda pollutant concentrations.

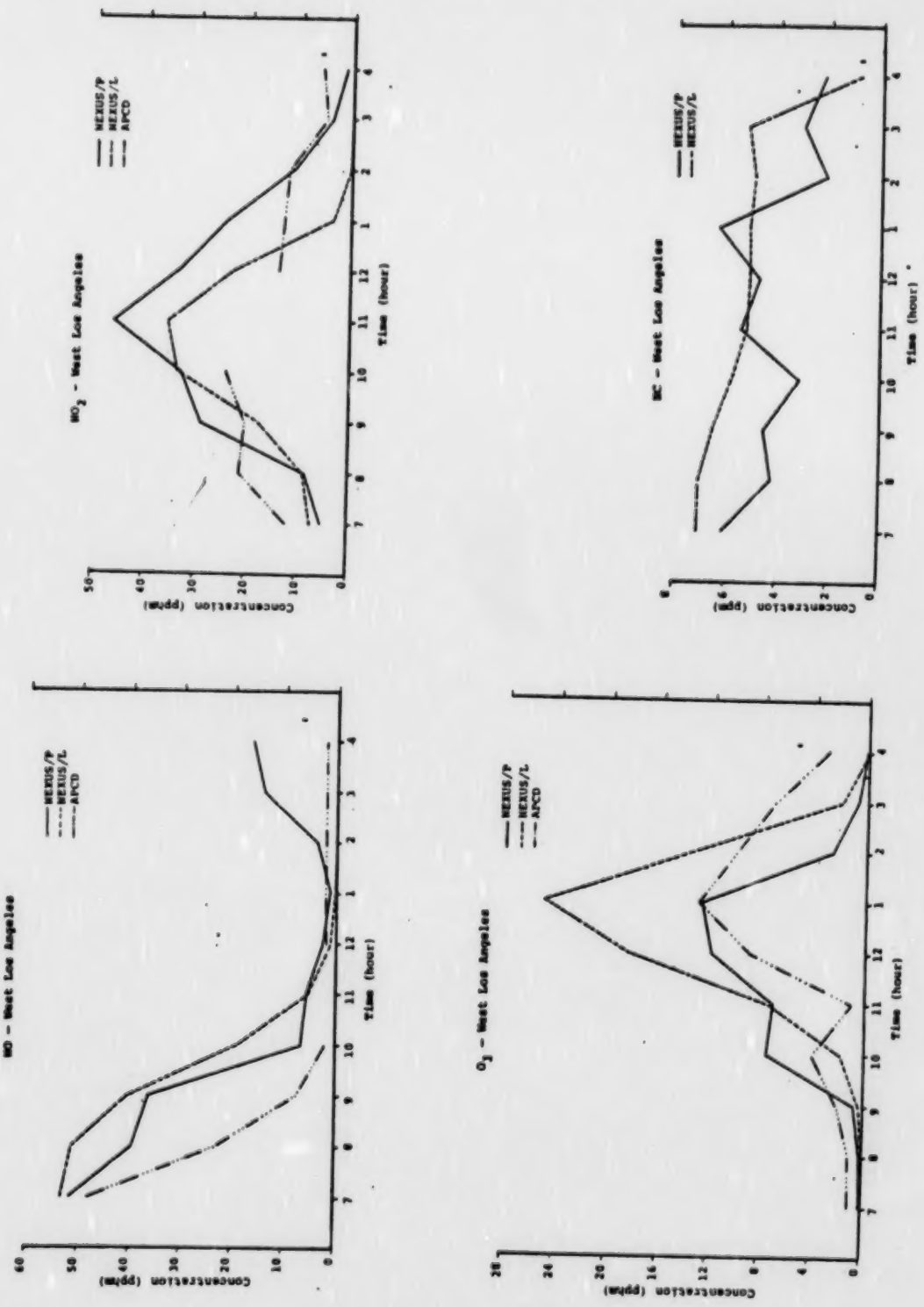


Figure 16. West Los Angeles pollutant concentrations.

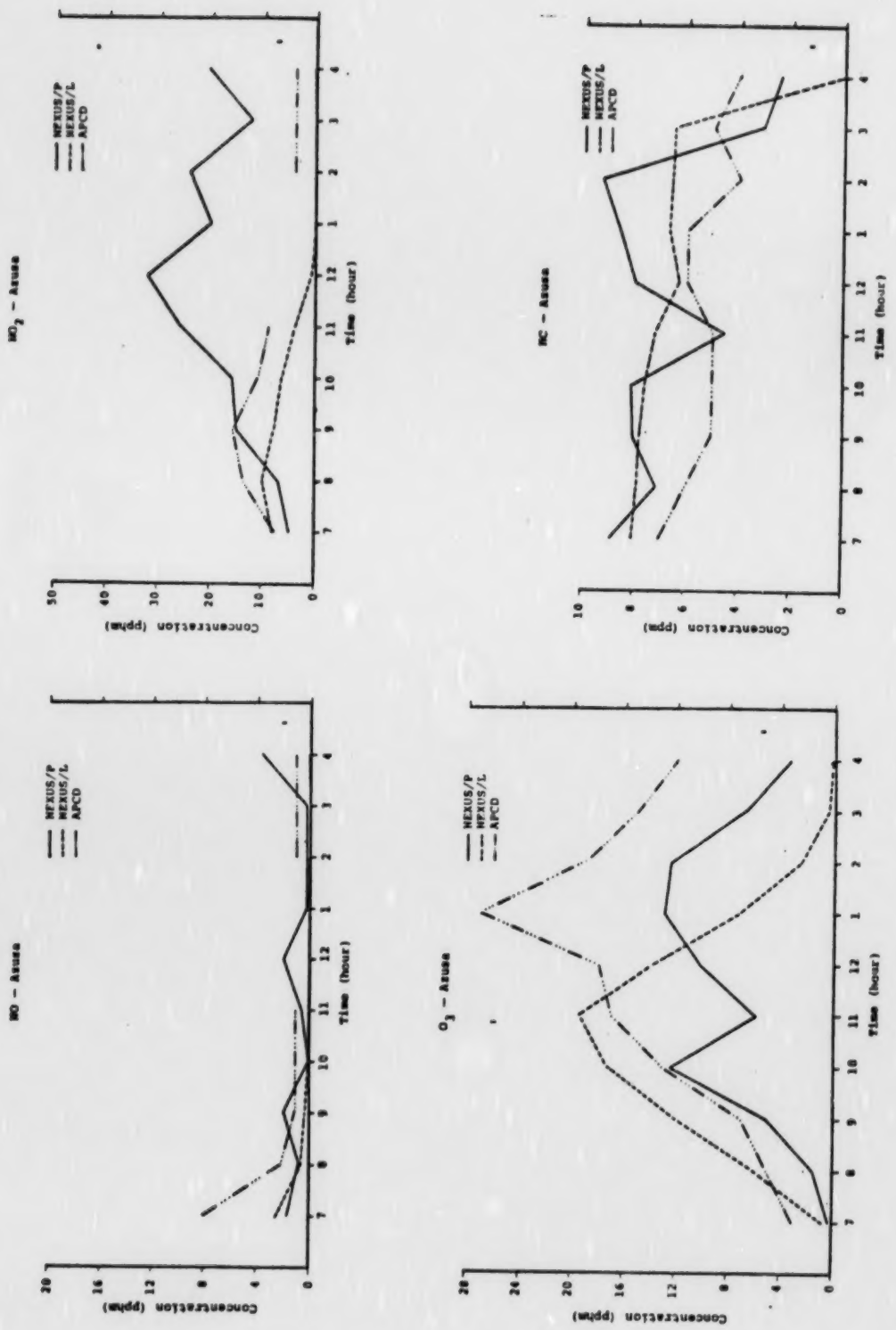


Figure 17. Azusa pollutant concentrations.

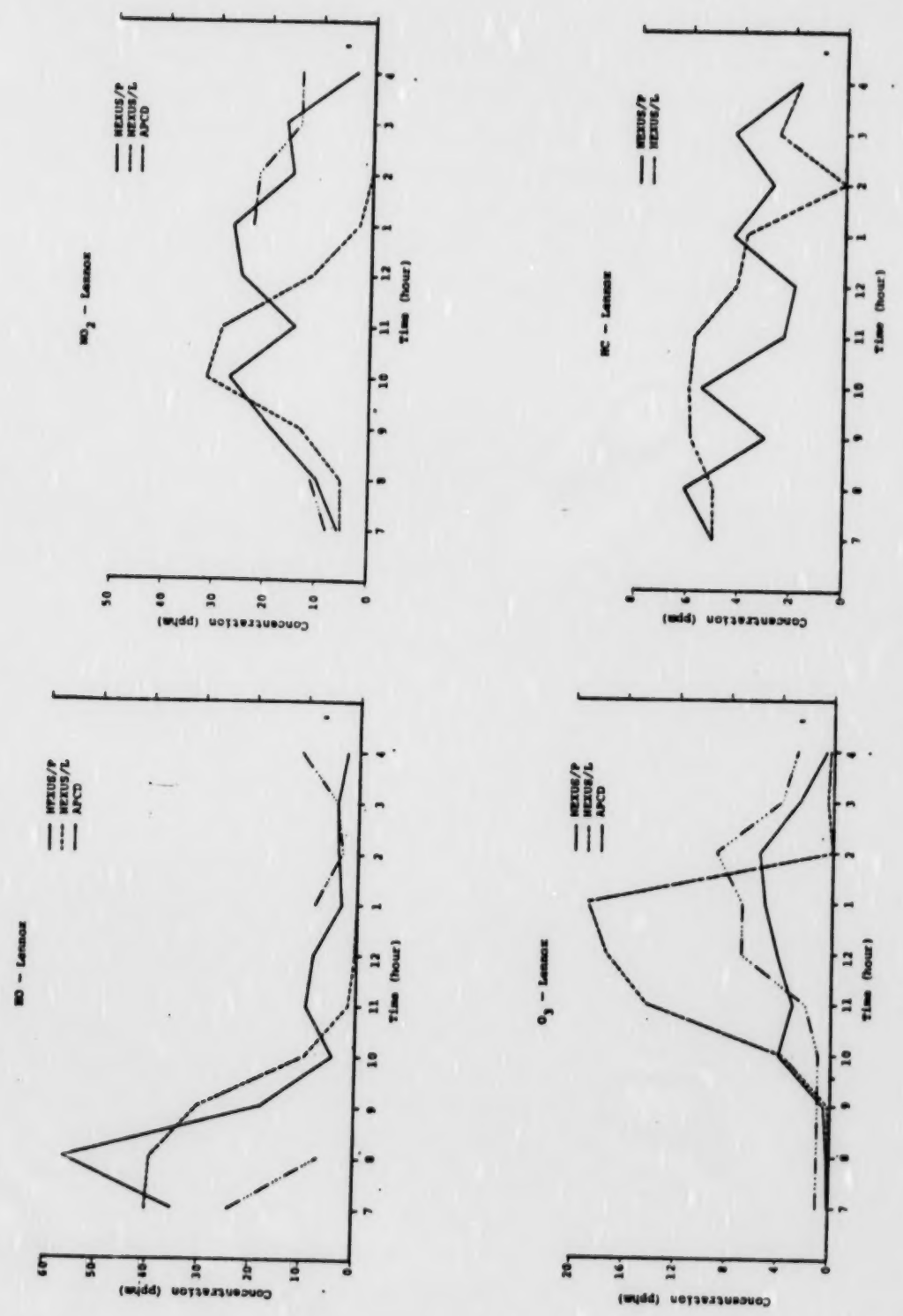


Figure 18. Lennox pollutant concentrations.

The general character of the NEXUS/L results is that they are much smoother than the NEXUS/P results. One reason for this is the averaging technique used to relate particle concentrations to cell concentrations for editing purposes. The general smoothing of the results is also indicated by examining the peak O_3 concentrations at each station. These results are shown in Table III. The NEXUS/L results over-predict at the stations near the coastline and underpredict in the high oxidant areas of Pasadena and Azusa.

TABLE III

PEAK OXIDANT CONCENTRATION

STATION	NEXUS/P	NEXUS/L	APCD
Pasadena	18	20	36
Long Beach	18	20	8
Lennox	4	18	8
West Los Angeles	12	26	12
Whittier	12	24	9
Azusa	12	20	28
Downtown	22	22	22
Burbank	21	24	20
Reseda	2	16	8

An examination of the comparison of the two different simulation results and the actual measurements indicate no trends which would allow one to choose one simulation as being superior to the other. From the results it is possible to determine that inherent errors in the Eulerian photochemistry are not so large that they cannot be overcome by adjusting the chemical reaction parameter. The examination of the chemical reaction system prior to the simulations to determine the rate

constants and branching factors has shown that the system is extremely sensitive to these parameters. The results of the calculations can be significantly affected by rather minor changes in these parameters. Therefore, the parameters themselves are the controlling factors in any simulation, not the simulation technique.

This draws one to the conclusion, that either Eulerian or Lagrangian chemistry is satisfactory given the current state-of-the-art in photochemical lump-parameter modeling.

A second photochemical simulation was run for Los Angeles for the same day using the NEXUS/L code. The purpose of this simulation was to test the sensitivity of the results to the input data. For the second simulation the sources were doubled so that twice as much total emissions were introduced into the problem. Some comparative results are shown in Figures 19 through 21. The calculated results do show a change in the general behavior of the pollutant concentrations. In particular, the NO_2 peak is much increased. However, the ozone peak is not increased particularly but is moved later in time. The same trend is noted at the Burbank and Downtown Los Angeles locations.

These results point out that a photochemical model such as NEXUS/L can be used for sensitivity calculations and will show differences because of changes in emissions.

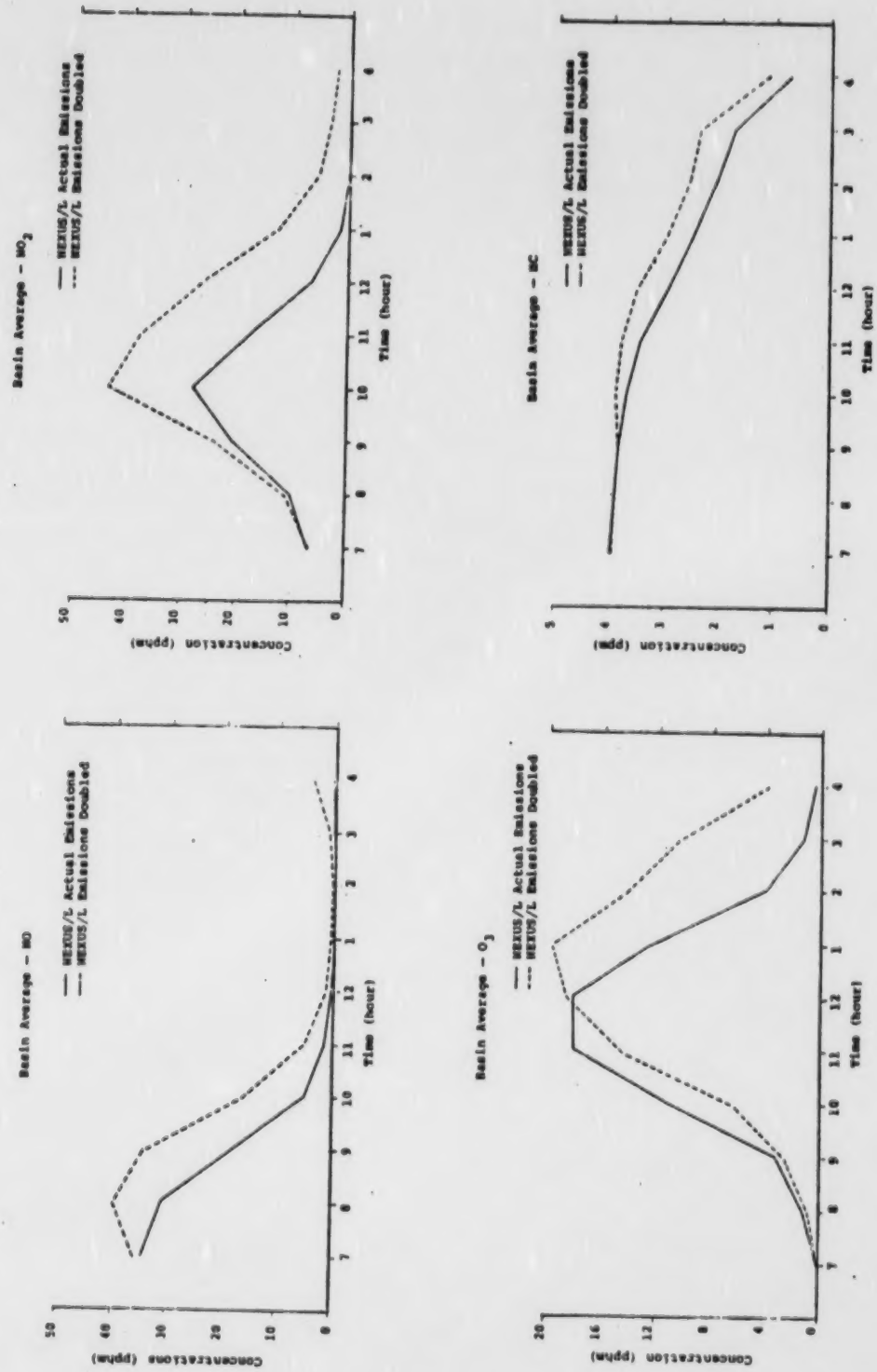


Figure 19. Comparison of basin-average NEXUS/L results using actual emissions and double emission.

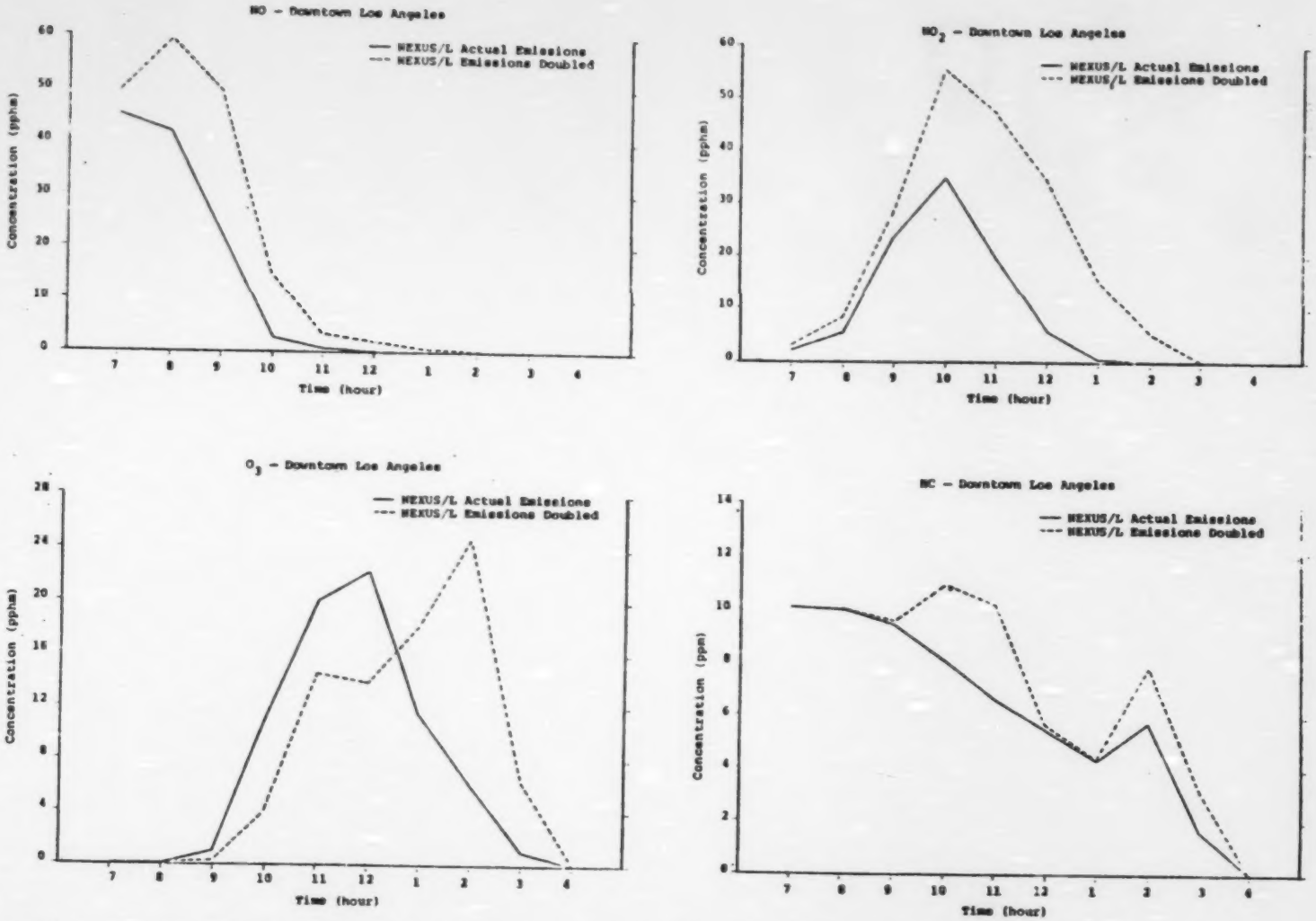


Figure 20. Comparison of Downtown Los Angeles NEXUS/L results for actual emissions and double emissions.

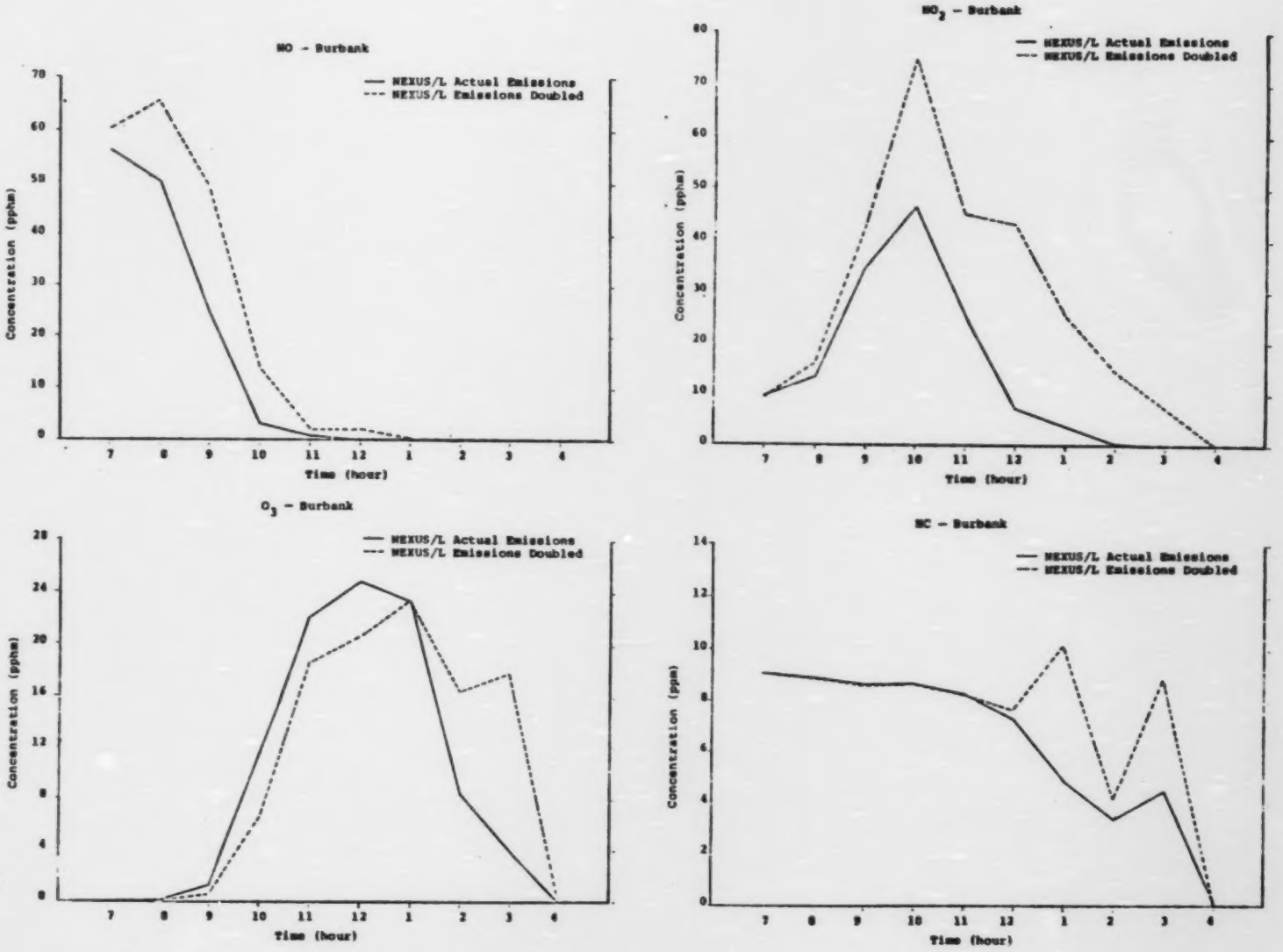


Figure 21. Comparison of Burbank NEXUS/L results for actual emissions and double emissions.

5. SUMMARY AND CONCLUSIONS

A new particle-in-cell code has been developed which employs Lagrangian chemistry and applied to the simulation of photochemical air pollution in the Los Angeles basin. These results are compared with NEXUS/P results and APCD measurements for the selected day. The results of the calculations indicate that either approach, the Eulerian or Lagrangian chemistry, can be used.

An additional calculation was made doubling all of the emissions. The oxidant levels as computed by the simulation were not significantly greater. This is consistent with the expectation that oxidant levels are effected by the NO_x/HC ratio rather than the actual emission levels. These results indicate that care must be taken in extrapolating air quality levels from change in emissions.

REFERENCES

1. Dickson, C.R. and G. Start, private communication (1971).

2. Eschenroeder, A.Q. and J.R. Martinez, "Concepts and Applications of Photochemical Smog Models," Report No. TM 1516 (June 1971), General Research Corporation, Santa Barbara, California.

3. Eschenroeder, A.Q. and J.R. Martinez, "Evaluation of a Photochemical Pollution Simulation Model," Contract No. 68-02-0336 (September 15, 1972), General Research Corporation, Santa Barbara, California.

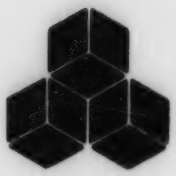
4. Roberts, P.J., P.M. Roth and C.L. Nelson, "Contaminant Models in the Los Angeles Basin – Their Sources Rates and Distribution," Report No. 71SAI-6 (1971), Systems Applications, Inc., Beverly Hills, California.

5. Roth, P.M., S.D. Reynolds, P.J. Roberts and J.H. Seinfeld, "Development of a Simulation Model for Estimating Ground Level Concentrations of Photochemical Pollutants," Report No. 71SAI-21 (July 1971), Systems Applications, Inc., Beverly Hills, California.

6. "Reactive Pollution Environment Simulation Model (REM) – User's Guide," (June 1970), Systems Development Corporation, Santa Monica, California.

7. "A Particle-In-Cell Method for Numerical Solution of the Atmospheric Diffusion Equations, and Applications to Air Pollution Problems," Contract No. 68-02-0006, Report No. 3SR-844 (November 1971), Systems, Science and Software, La Jolla, California.

8. "Mathematical Modeling of Photochemical Smog Using the PICK Method," Contract No. 68-02-0006, APCA Paper No. 72-140 (June 1972), Systems, Science and Software, La Jolla, California.



SYSTEMS, SCIENCE AND SOFTWARE

SSS-R-74-1756

PART II

OPERATIONS MANUAL

P.O. BOX 1620, LA JOLLA, CALIFORNIA 92037, TELEPHONE (714) 453-0060

1. INTRODUCTION

Systems, Science and Software (S^3) has previously developed and documented the computer codes NEXUS/P and SETUP.[1]

A new code, NEXUS/L, has been developed and tested. It differs from the NEXUS/P code in that the photochemical kinetics are considered in a Lagrangian framework, within which each particle contains a specific concentration of each pollutant specie being considered. Previously, in the NEXUS/P code, each particle represented a fixed mass of an inidividual pollutant.

The purpose of this volume is to document the NEXUS/L code and provide a users guide to its operation.

2. DISCUSSION OF THE NEXUS/L CODE

2.1 GRID LABELING CONVENTIONS

It is useful to be able to reference the Eulerian grid in both cell space (dimensionless) and measured space (meters). The principal axes are noted x, y, and z. The horizontal planes form x-y space, and z is the vertical coordinate.

The cells have dimensions DX, DY, and DZ, and the full grid has NX, NY, and NZ cells in the principal axis directions, respectively. Figure 1 shows a two-dimensional cross-section of the grid. The labeling conventions are the same as with the NEXUS/P code. In cell space, the cell centers are integer triplets [I,J,K] in the x, y, and z directions, respectively. The origin in cell space is [0.5, 0.5, 0.5]. In measured space, the origin is at [0, 0, 0] meters. The center of cell [I,J,K], defined in meters, is [(I-0.5)DX, (J-0.5)DY, (K-0.5)DZ].

2.2 THE NEXUS/L PARTICLES

Each pollutant specie has an individual concentration within each particle. The particles have arbitrarily been sized as exactly one Eulerian cell, DX, DY, DZ, in volume. The concentrations are attributed to the entire particle volume. The position of particle N is defined by the position of its center $[X_N, Y_N, Z_N]$. It is convenient to use cell space notation for particle position.

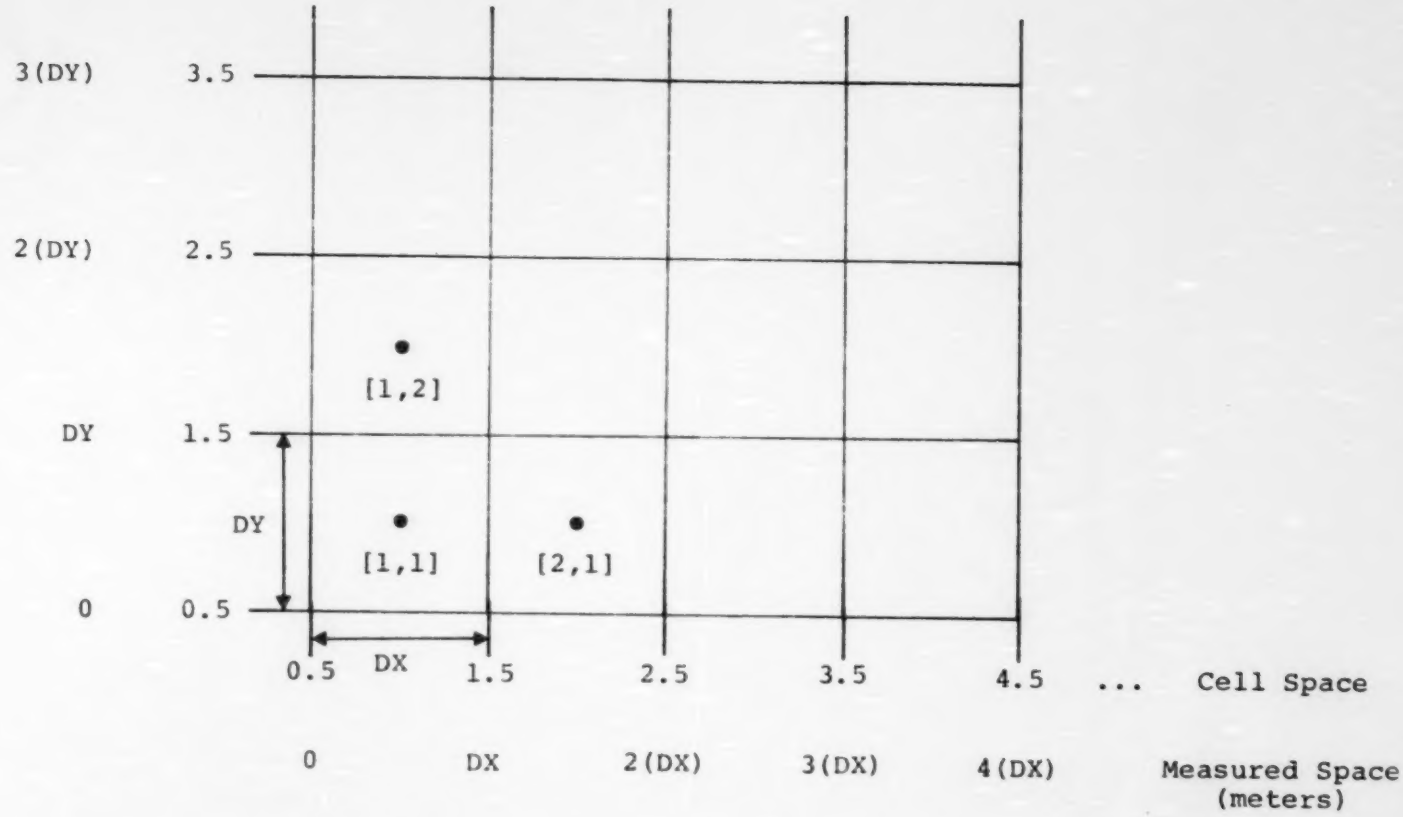


Figure 1. Grid labeling conventions.

SSS-R-74-1756

2.3 FLOW LOGIC OF THE NEXUS/L CODE

The calculational sequence of a typical cycle of the NEXUS/L code is as follows:

(1) advance the cycle counter

(2) obtain updated wind field, diffusivities, and source emission rates (Eulerian framework)

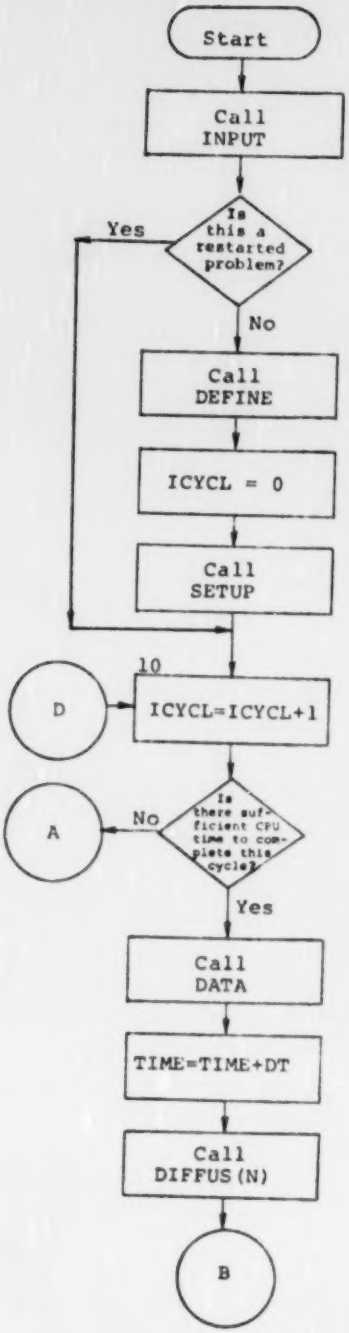
(3) advance the problem time

(4) calculate rate of change of concentration due to diffusion (Eulerian framework)

(5) update the particle concentrations to reflect the effects of source emissions and diffusion and photochemical reactions (Lagrangian framework)

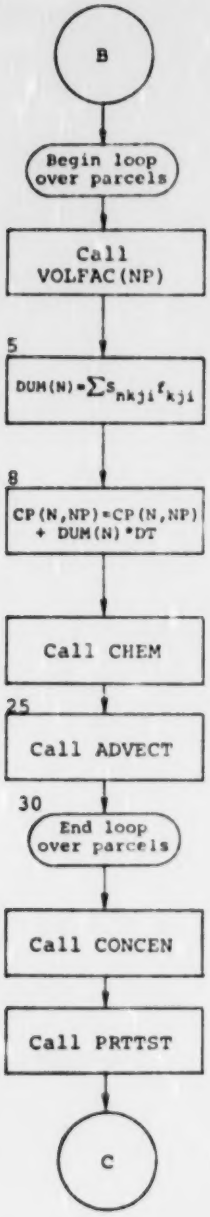
(6) advect the particles by the wind field to new positions (Lagrangian framework)

A flow chart of the NEXUS/L main program is given in Figures 2(a), (b), and (c).



- determine if problem is to be restarted or generated fresh. Set up all control parameters.
- set up certain indices and constants
- initialize cycle counter
- initialize location and concentration of Lagrangian parcels
- main loop, advance cycle counter
- obtain winds, diffusivities, and sources for this cycle in Eulerian framework
- advance the problem time
- obtain diffusion contribution of each specie in Eulerian framework

Figure 2(a). Flow logic of NEXUS/L main program.



- obtain fractional volumes of this parcel with respect to the Eulerian framework
- use fractional volumes to develop the parcel source term for each specie as a function of the Eulerian sources
- the concentration of each specie is adjusted to reflect the diffusion and new source emissions contributions
- the effects of photochemical reactions are incorporated
- the parcel is advected by the winds to a new coordinate position
- relate concentrations in the Lagrangian parcels to the Eulerian grid framework
- determine if this cycle's results are to be printed, plotted, or saved on a dump file

Figure 2(b). Flow logic of NEXUS/L main program.

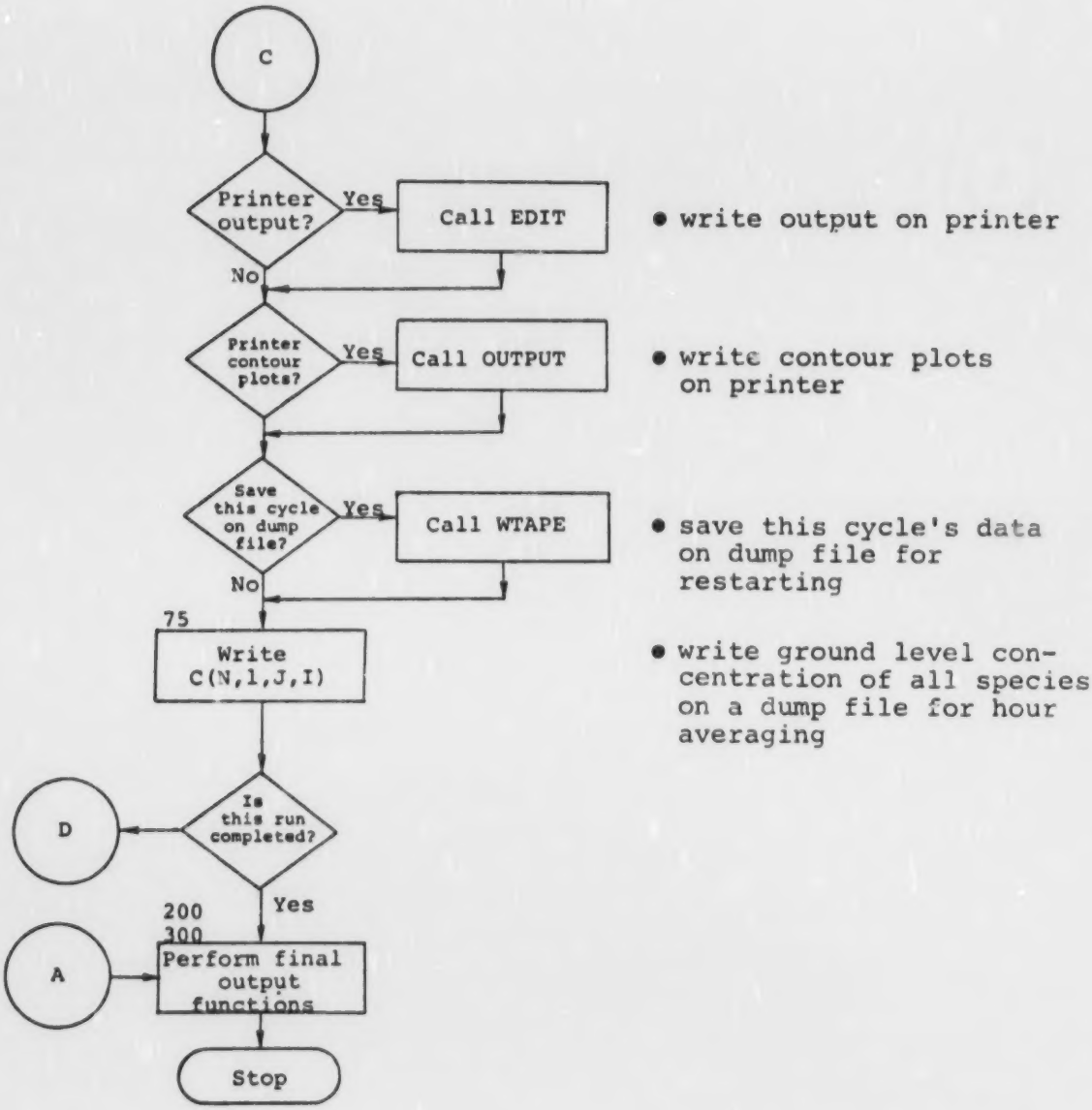


Figure 2(c). Flow logic of NEXUS/L main program.

2.4 GLOSSARY OF KEY QUANTITIES

Quantity	Description	Units
DX	x-direction cell width	m
DY	y-direction cell width	m
DZ	z-direction cell width	m
NX	number of cells in x-direction	
NY	number of cells in y-direction	
NZ	number of cells in z-direction	
NPM	number of Lagrangian particles	
NSP	number of pollutant species	
ICYCL	current problem cycle number	
TIME	current problem time	sec
DT	current cycle time step	sec
U(I,J,K)	x-direction velocity at center of cell i,j,k	m/sec
V(I,J,K)	y-direction velocity at center of cell i,j,k	m/sec
W(I,J,K)	z-direction velocity at center of cell i,j,k	m/sec
EX(I,J,K)	eddy diffusivity for x- and y-direction at center of cell i,j,k	m^2/sec
EZ(I,J,K)	eddy diffusivity for z-direction at center of cell i,j,k	m^2/sec
S(NSP,K,J,I)	specie source rate at center of cell i,j,k	PPM/sec
C(NSP,K,J,I)	specie concentration in Eulerian cell i,j,k	PPM
X(NP)	x-coordinate of particle NP in cell space	
Y(NP)	y-coordinate of particle NP in cell space	
Z(NP)	z-coordinate of particle NP in cell space	
CP(NSP,NP)	specie concentration in particle NP	PPM

2.5 HOW PROBLEMS ARE GENERATED

Subroutine INPUT directs the initialization of calculation. A flow chart of this subroutine appears in Figure 3. Code variables designated "not currently used" exist to permit easy implementation of new code options without major changes to BLANK COMMON and/or subroutine INPUT.

2.5.1 Namelist "START"

"START" is used to provide an overview of the problem to be computed.

RESTRT: If zero, the problem is to be generated new;

If > 0, the problem is to be restarted and continued.

ISTART: Defines the cycle number at which the problem is to be restarted.

TMAX: A time (in seconds) which, if exceeded in the calculational time, the calculation is to stop.

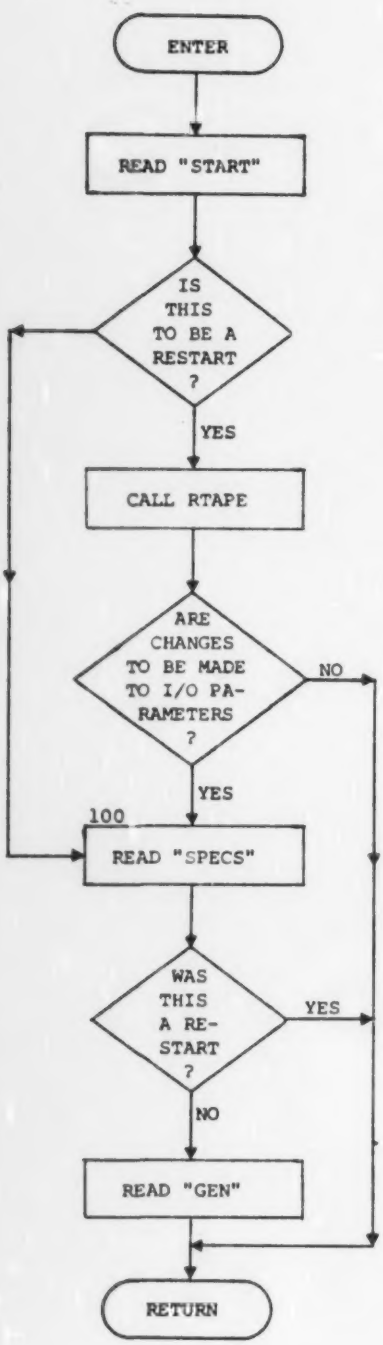
CYCMAX: A cycle number, which, if exceeded in the calculational process, the calculation is to stop.

CHANGE: If zero, has no effect;

If > 0, and the problem is a restart, requests the variables in namelist "SPECS" to be redefined (see 2.5.2 below).

2.5.2 Namelist "SPECS"

"SPECS" is used to define output control parameters, printer edits, printer contour plots, and dumps on permanent files.



- read data in namelist "START"
- space through restart data file and read data for appropriate start cycle
- read data in namelist "SPECS"
- read data in namelist "GEN"

Figure 3. Flow logic of subroutine INPUT.

IDUMP: Controls dumps of blank common.

If non-zero, dumps will be performed whenever MOD(ICYCL,IDUMP) is zero.

TDUMP: Permits control of dumps on problem calculational time instead of cycle number.

If non-zero, dumps will be performed each time k*DUMP exceeds TIME.

Note: after each such dump, the integer k is incremented by 1.

IPRINT: Controls printer edits. Works like IDUMP.

TPRINT: Works like TDUMP for printer edits.

IPLOT: Controls printer contour plots. Works like IDUMP.

TPLOT: Controls printer contour plots. Works like TDUMP.

LEVELS: Used to shorten printer edits, if desired. The z-levels edited are 1-LEVELS rather than 1-NZ.

I1,I2,I3,I4,I5: Not currently used.

NXY: Number of x-y plane contour plots desired (maximum of 3).

NXZ: Number of x-z plane contour plots desired (maximum of 3).

NYZ: Number of y-z plane contour plots desired.

XYP(3): Used to specify the z-cross sections desired for the (NXY) x-y plane contour plots.

XZP(3): Used to specify the y-cross sections desired for the (NXZ) x-z plane contour plots.

YZP(3): Used to specify the x-cross sections desired for the (NYZ) y-z plane contour plots.

ISP(5): Used to choose which species are to be contour plotted.

If ISP(K) = 0, specie k is not plotted.

If ISP(K) > 0, specie k is available for plotting.

2.5.3 Namelist "GEN"

"GEN" is used to define the grid characteristics, and in the case of certain test problems, "GEN" can be used to define the winds and diffusivities.

NX: Number of x-direction cells.

NY: Number of y-direction cells.

NZ: Number of z-direction cells.

NSP: Number of pollutant species.

DX: Width of a cell in x-direction.

DY: Width of a cell in y-direction.

DZ: Width of a cell in z-direction.

NPC: Number of Lagrangian parcels to be created for each Eulerian grid cell.

NS: Not currently used.

TIME: Calculation time (sec) at which the problem is to be initialized.

DT: Calculational time step to be used.

FLAG1, FLAG2, IFLAG1, IFLAG2: Not currently used.

CONST: (Logical Variable) Used to quickly set up test problems.

If input as .TRUE., then winds and diffusivities will be constant throughout the calculation:

The following are input if CONST = .TRUE.

UFIX: u-velocity in each cell

VFIX: v-velocity in each cell

WFIX: w-velocity in each cell

EXFIX: x,y diffusivities in each cell

EZFIX: z diffusivity in each cell

GROUND: (Logical Variable) Set to .TRUE. if it is desired to dump ground level concentrations on a permanent file each cycle.

3. CODE LISTING

The NEXUS/L code, as used to run the Los Angeles photochemical simulation, is documented below. The code elements are in the following order:

procedure	BLANK
subroutine	ADVECT
subroutine	CHEM
subroutine	CONCEN
subroutine	CONTUR
subroutine	DATA
subroutine	DEFINE
subroutine	DIFFUS
subroutine	EDIT
subroutine	INPUT
program	MAIN
subroutine	OUTPUT
subroutine	PL
subroutine	PRTTST
subroutine	RTAPE
subroutine	SETUP
subroutine	VOLFAC
subroutine	WTAPE

Reproduced from
best available copy.

BLANK COMMON

```
      .BLANK
      BLANK  PROC                                                       BLANK001
C                                                                       BLANK002
C.....PARAMETER STATEMENTS                                              BLANK003
C                                                                       BLANK004
      PARAMETER (NXI=22,NYI=21,NZI=4,NSPI=5,MAX=2772                    BLANK005
     1 PARAMETER (NUMNSP)=(NXI)*(NYI)*(NZI+LUMNXI)*(NYI)*(NZI           BLANK006
C.....                                                                  BLANK007
C                                                                       BLANK008
C.....BLANK COMMON                                                      BLANK009
      COMMON GUM(101),IDUM(101),NX,NY,NZ,NSP,NPC,NP,NS,NPM,NCOMM,      BLANK010
     1  NXP1,NYP1,NZP1,NYM1,NZM1,IP1,JP1,KP1,IM1,JM1,KM1,               BLANK011
     2  TIME,DT,TMAX,CYCMAX,SAVE,RESTRT,ICYCL,PRINTT,PLOT,DUMP,GROUND   BLANK012
     3  ISTART,IPRINT,IPRINT,IPLOT,IPLOT,IDUMP,IDUMP,TIMP,TIMP,         BLANK013
     4  TIMO,NTAPE,PLAG1,PLAG2,IFLAG1,IFLAG2,I1,I2,I3,I4,I5,ALLVELS,    BLANK014
     5  DX,DY,DZ,DTDX,DTDY,DTDZ,RDX,RDY,RDZ,RDX2,RDY2,RDZ2,DUMMY,CONS   BLANK015
     6  U(NXI,NYI,NZI),V(NXI,NYI,NZI),W(NXI,NYI,NZI),CP(NSPI,MAX),      BLANK016
     7  CONSP(NXI,NYI,NXI),S(NSPI,NZI,NYI,NXI),X(MAX),Y(MAX),Z(MAX),    BLANK017
     8  EE(NXI,NYI,NZI),F2(NXI,NYI,NZI),UFIX,VFIX,WFIX,RFIX,EXFIX,EZFIX BLANK018
C                                                                       BLANK019
      COMMON /IO/ NXY,NXZ,NYZ,XYP(3),XZP(3),YZP(3),ISP(5),TITLE(15)     BLANK020
C                                                                       BLANK021
      LOGICAL PLOT,PRINTT,DUMP,SAVE,CONST,GROUND                        BLANK022
      INTEGER DUMMY                                                     BLANK023
C                                                                       BLANK024
      END                                                               BLANK025
```

```
      .ADVECT
      SUBROUTINE ADVECT                                                 ADVECT01
      INCLUDE BLANK                                                     ADVECT02
      COMMON /PANCEL/ II,JJ,KK,F1,F2,F3,F4,F5,F6,F7,F8,I,J,K           ADVECT03
C                                                                       ADVECT04
C                                                                       ADVECT05
C                                                                       ADVECT06
      V1= U(I,J,K)*F1+U(II,J,K)*F2+U(II,JJ,K)*F3+U(I,JJ,K)*F4 +         ADVECT07
     1    U(I,J,KK)*F5+U(II,J,KK)*F6+U(II,JJ,KK)*F7+U(I,JJ,KK)*F8      ADVECT08
      V2= V(I,J,K)*F1+V(II,J,K)*F2+V(II,JJ,K)*F3+V(I,JJ,K)*F4 +         ADVECT09
     2    V(I,J,KK)*F5+V(II,J,KK)*F6+V(II,JJ,KK)*F7+V(I,JJ,KK)*F8      ADVECT10
C**********SPECIAL DEAL FOR THE BOTTOM LEVEL                           ADVECT11
      IF(Z(NP) .LT. 1.5) GO TO 9                                        ADVECT12
      V3= W(I,J,K)*F1+W(II,J,K)*F2+W(II,JJ,K)*F3+W(I,JJ,K)*F4 +         ADVECT13
     3    W(I,J,KK)*F5+W(II,J,KK)*F6+W(II,JJ,KK)*F7+W(I,JJ,KK)*F8      ADVECT14
      GO TO 8                                                           ADVECT15
    9 V3=(Z(NP)-0.5)*0.5*((F1+F5)*W(I,J,2)+(F2+F6)*W(II,J,2)            ADVECT16
     1   +(F3+F7)*W(I,JJ,2)+(F4+F8)*W(I,JJ,2))                        ADVECT17
C                                                                       ADVECT18
C                                                                       ADVECT19
    8 X(NP)= X(NP) + V1 * DTDX                                          ADVECT20
      Y(NP)= Y(NP) + V2 * DTDY                                          ADVECT21
      Z(NP)= Z(NP) + V3 * DTDZ                                          ADVECT22
C                                                  TEMPORARY B.C.       ADVECT23
      ZERO=0.0                                                          ADVECT24
      IF(X(NP) .GE. 0.5) GO TO 10                                       ADVECT25
C     **X < 0.5                                                         ADVECT26
      X(NP)=X(NP)+NX                                                    ADVECT27
      ZERO=1.0                                                          ADVECT28
   10 IF(X(NP) .LT. NX+0.5) GO TO 20                                    ADVECT29
C     **X > NX+0.5                                                      ADVECT30
      X(NP)=X(NP)-NX                                                    ADVECT31
      ZERO=1.0                                                          ADVECT32
   20 IF(Y(NP) .GE. 0.5) GO TO 30                                       ADVECT33
C     **Y < 0.5                                                         ADVECT34
      Y(NP)=Y(NP)+NY                                                    ADVECT35
      ZERO=1.0                                                          ADVECT36
   30 IF(Y(NP) .LT. NY+0.5) GO TO 40                                    ADVECT37
C     **Y > NY+0.5                                                      ADVECT38
      Y(NP)=Y(NP)-NY                                                    ADVECT39
      ZERO=1.0                                                          ADVECT40
   40 IF(Z(NP) .GE. 0.5) GO TO 50                                       ADVECT41
C     **Z < 0.5                                                         ADVECT42
      Z(NP)=.501                                                        ADVECT43
   50 IF(Z(NP) .LT. NZ+0.5) GO TO 100                                   ADVECT44
      Z(NP)=Z(NP)-NZ                                                    ADVECT45
      ZERO=1.0                                                          ADVECT46
C                                                                       ADVECT47
  100 IF(ZERO .LT. 0.5) GO TO 200                                       ADVECT48
      DO 150 N=1,NSP                                                    ADVECT49
      CP(N,NP)=0.0                                                      ADVECT50
  150 CONTINUE                                                          ADVECT51
C                                                                       ADVECT52
  200 RETURN                                                            ADVECT53
```

```
ADVECT
      SN      C
      SS      C      END

      ADVECTSN
      ADVECTSS
```

```
ANSCHK
      1          .ANSCHK
      2                SUBROUTINE ANSCHK(RR,RY)
      3                LOGICAL RY
      4                RY=.FALSE.
      5                IF(RR .EQ. 'YES   ') GO TO 50
      6                IF(RR .EQ. 'YES  ') GO TO 50
      7                IF(RR .EQ. 'YES ') GO TO 50
      8                IF(RR .EQ. 'YES') GO TO 50
      9                GO TO 100
     10           50 RY=.TRUE.
     11        C
     12          100 RETURN
     13        C
               END
```

CHEM

```
      .CHEM
      SUBROUTINE CHEM(ONO,ONO2,OO3,OHNO2,OHC,DUM1,DUM2,DUM3,DUM4,DUM5,   CHEM0001
     1   DELTAT,TIM)                                                     CHEM0002
C     THIS ROUTINE SOLVES THE CHEMICAL RATE CHANGE EQUATIONS AND RETUR SCHEM0003
C     THE FRACTIONAL CHANGE IN CONCENTRATION OF EACH SPECIE.            CHEM0004
      DATA CONE, C1A, C2, C3, C4, C5, C6, C7, C8, C9, CTEN, C11, C12,   CHEM0005
     1 C13, C14, C15 / .267, 2.69E6, 26.7, 2.81E-5, 1.5E4, .004,       CHEM0006
     2 1.E5, 200., 1500., 3000., .001, .01, .05, 4500., 14., 60.5 /    CHEM0007
      DATA B1, B2, B3, B4 / 2*10., 2*1/                                 CHEM0008
      REAL NO, NO2, NO3, N2O5                                           CHEM0009
      DIMENSION Y(10), YS(10), A(5), B(5), R(5), E(3), LF(3)            CHEM0010
      DATA FR / .625 /                                                  CHEM0011
      EQUIVALENCE (Y(1),NO), (Y(2),NO2), (Y(3),O3), (Y(4),HNO2),        CHEM0012
     1 (Y(5),HC), (Y(6),O), (Y(7),OH),  (Y(8),RO2), (Y(9),NO3),         CHEM0013
     2 (Y(10),N2O5)                                                     CHEM0014
      NAMELIST /DUDIFY / C1A,C2,C3,C4,C5,C6,C7,C8,C9,CTEN,C11,C12,C13,  CHEM0015
     1   C14,C15,B1,B2,B3,B4,FR                                         CHEM0016
C                                                                       CHEM0017
      DEFINE F1 = C6*NO + C7*NO2                                        CHEM0018
      DEFINE F2 = C6*B4*NO                                              CHEM0019
      DEFINE F3 = C14 + C15                                             CHEM0020
      DEFINE F4 = C4*HC + CR*NO + C9*NO2                                CHEM0021
      DEFINE F5 = C6*B4*RO2*NO + C10*HNO2                               CHEM0022
      DEFINE FNO = C1*NO2 + C10*HNO2                                    CHEM0023
      DEFINE RNO = C2*O3 + C6*RO2 + C8*OH + C11*NO2                     CHEM0024
      DEFINE FNO2 = 2*C2*O3 + C6*RO2*NO + C14*N2O5                      CHEM0025
      DEFINE RNO2 = C1 + C7*RO2 + C9*OH + C11*NO + C12*O3 + C13*NO3     CHEM0026
      DEFINE F03 = C1A*O                                                CHEM0027
      DEFINE RO3 = C2*NO + C5*HC + C12*NO2                              CHEM0028
      DEFINE FHNO2 = C8*OH*RO + 2.*C11*NO*NO2                           CHEM0029
      DEFINE RHNO2 = C10                                                CHEM0030
      DEFINE RHC = C3*O + C4*OH + C5*O3                                 CHEM0031
      DEFINE EO = C1*NO2/(C1A + C3*HC)                                  CHEM0032
      DEFINE EOH = (F2*C3*B1*O*HC + F2*C5*B3*O3*HC + F1*C10*HNO2)/      CHEM0033
     1 (F1*F4 + F2*C4*B2*HC)                                            CHEM0034
      DEFINE ERO2 = HC*(C3*B1*O + C4*B2*OH + C5*B3*O3)/F1               CHEM0035
      DEFINE ENO3 = C12*O3*F3/(C13*C15)                                 CHEM0036
      DEFINE EN2O5 = C13*NO2*NO3/F3                                     CHEM0037
      DEFINE RATE(I) = (A(I) - B(I)*Y(I))*TT                            CHEM0038
C                                                                       CHEM0039
C                                                                       CHEM0040
    3 ITEST=0                                                           CHEM0041
      TIME = 0.                                                         CHEM0042
C     PHOTOCHEMICAL REACTION RATE VARIES SINUSOIDALLY FROM 6 AM TO 6 P  CHEM0043
      TIMH=TIM/3600.                                                    CHEM0044
    5 DT = 1.0E-6                                                       CHEM0045
      TT=DELTAT/60.0                                                    CHEM0046
      NO = ONO                                                          CHEM0047
      NO2 = ONO2                                                        CHEM0048
      HC = OHC                                                          CHEM0049
      O3 = OO3                                                          CHEM0050
      HNO2 = OHNO2                                                      CHEM0051
   10 NT = 1                                                            CHEM0052
      SINC = SIN((TIMH - 6. - TT/60. + TIME/60.)*.2618)                 CHEM0053
```

```
      IF(SINC .LT. 0.0) SINC=0.0                                        CHEM0054
      C1 = CONE*SINC                                                    CHEM0055
      C10 = CTEN*SINC                                                   CHEM0056
   12 O = EO                                                            CHEM0057
      OH = EOH                                                          CHEM0058
      RO2 = ENO2                                                        CHEM0059
      NO3 = ENO3                                                        CHEM0060
      N2O5 = EN2O5                                                      CHEM0061
C     **ADD SOURCE RATES TO FORMATION RATES                             CHEM0062
      A(1)=FNO+DUM1                                                     CHEM0063
      A(2)=FNO2+DUM2                                                    CHEM0064
      A(3)=FO3+DUM3                                                     CHEM0065
      A(4)=FHNO2+DUM4                                                   CHEM0066
      A(5)=DUM5                                                         CHEM0067
      B(1) = RNO                                                        CHEM0068
      B(2) = RNO2                                                       CHEM0069
      B(3) = RO3                                                        CHEM0070
      B(4) = RHNO2                                                      CHEM0071
      B(5) = RHC                                                        CHEM0072
      IF (NT .EQ. 2) GO TO 38                                           CHEM0073
      NT = 2                                                            CHEM0074
C     TEST TO SEE IF SIGNIFICANT PHOTOCHEMICAL REACTION WILL OCCUR THI  CHEM0075
C     TIME STEP, AND. IF NOT, USE SIMPLE RATE EQUATION AND RETURN       CHEM0076
      IF (ITEST .NE. 0) GO TO 19                                        CHEM0077
      ITEST = 1                                                         CHEM0078
      DO 15 I=1,5                                                       CHEM0079
      R(I)=RATE(I)                                                      CHEM0080
   15 CONTINUE                                                          CHEM0081
      XTEST =  ABS((Y(1) + R(1)   )/(Y(2) + R(2)   )) / ABS((Y(1)/Y(2)) CHEM0082
      IF(ABS(R(1)) .LT. 1.0E-10  .AND.  ABS(R(2)) .LT. 1.0E-10) XTEST= CHEM0083
      IF (ABS(R(3)) .GT. .01 .OR. XTEST .GT. 1.1 .OR. XTEST .LT. .9)    CHEM0084
     1 GO TO 19                                                         CHEM0085
      DO 16 I = 1, 3                                                    CHEM0086
      LF(I) = 0                                                         CHEM0087
      E(I) = A(I)/B(I)                                                  CHEM0088
      IF (Y(I) .LT. E(I)) LF(I) = 1                                     CHEM0089
   16 CONTINUE                                                          CHEM0090
      DO 17 I=1,5                                                       CHEM0091
      Q1=EXP(-B(I)*DT)                                                  CHEM0092
      QTEST=(1.0-Q1)*A(I)/B(I)+Q1*Y(I)                                  CHEM0093
      Y(I)=MAX(0.0,QTEST)                                               CHEM0094
   17 CONTINUE                                                          CHEM0095
      E(3) = FO3/RO3                                                    CHEM0096
      DO 18 I = 1, 3                                                    CHEM0097
      IF (Y(I) .LT. E(I) .AND. LF(I) .EQ. 0 .OR. Y(I) .GT. E(I) .AND.   CHEM0098
     1 LF(I) .EQ. 1) Y(I) = E(I)                                        CHEM0099
   18 CONTINUE                                                          CHEM0100
      XTEST = (Y(1)/Y(2))/(ONO/ONO2)                                    CHEM0101
      IF(XTEST.GT.0.9.AND.XTEST.LT.1.1) GO TO 900                       CHEM0102
      GO TO 5                                                           CHEM0103
   19 CONTINUE                                                          CHEM0104
      DO 30 I = 1, 5                                                    CHEM0105
      YS(I) = Y(I)                                                      CHEM0106
      IF (B(I)*DT .GT. 1.E-7) GO TO 20                                  CHEM0107
```

```
 CHEM
      108            Y(I) = .5*A(I)*DT + Y(I)                                                    CHEM0108
      109            GO TO 30                                                                    CHEM0109
      110         20 Y(I) = (A(I)/B(I)) + (Y(I) - A(I)/B(I))*EXP(-B(I)*DT) + Y(I))*.5            CHEM0110
      111         30 CONTINUE                                                                    CHEM0111
      112            GO TO 12                                                                    CHEM0112
      113         38 DO 50 I = 1, 5                                                              CHEM0113
      114            IF (A(I)*DT .GT. 1.E-7) GO TO 40                                            CHEM0114
      115            Y(I) = A(I)*DT + YS(I)                                                      CHEM0115
      116            GO TO 50                                                                    CHEM0116
      117         40 Y(I) = A(I)/B(I) + (YS(I) - A(I)/B(I))*EXP(-B(I)*DT)                        CHEM0117
      118         50 CONTINUE                                                                    CHEM0118
      119            TIME = TIME + DT                                                            CHEM0119
      120            IF (TIME .GE. TT) GO TO 900                                                 CHEM0120
      121            C = 0.                                                                      CHEM0121
      122            DO 60 I = 1, 5                                                              CHEM0122
      123            IF (Y(I) .LT. 1.E-9) GO TO 60                                               CHEM0123
      124            C = MAX(C, ABS((Y(I) - YS(I))/(Y(I) + YS(I))))                              CHEM0124
      125         60 CONTINUE                                                                    CHEM0125
      126            C = MAX(C, .1*ER)                                                           CHEM0126
      127            DT = MIN(ER/C*DT, TT - TIME)                                                CHEM0127
      128            GO TO 10                                                                    CHEM0128
      129      C*****************************************************************************  CHEM0129
      130      C     *UPDATE CP(*) ARRAY AND RETURN TO MAIN PROGRAM                          *  CHEM0130
      131        900 CALL MO                                                                     CHEM0131
      132            CNO2=PO2                                                                    CHEM0132
      133            CHC=HC                                                                      CHEM0133
      134            CO3=O3                                                                      CHEM0134
      135            CHNO2=HNO2                                                                  CHEM0135
      136            RETURN                                                                      CHEM0136
      137            END                                                                         CHEM0137
```

CONCEN

```
      .CONCEN
      SUBROUTINE CONCEN                                                 CONCEN01
      INCLUDE BLANK                                                     CONCEN02
      COMMON /PARCEL/ II,JJ,KK,F1,F2,F3,F4,F5,F6,F7,F8,I,J,K           CONCEN03
C                                                                       CONCEN04
C     **ZERO OUT C,S ARRAYS                                             CONCEN05
C                                                                       CONCEN06
      CALL S3ZERO(C,NUM)                                                CONCEN07
      CALL S3ZERO(S,NUM)                                                CONCEN08
C                                                                       CONCEN09
      DO 100 NP=1,NPH                                                   CONCEN10
C                                                                       CONCEN11
      CALL VOLFAC(NP)                                                   CONCEN12
C                                                                       CONCEN13
      DO 50 N=1,NSP                                                     CONCEN14
C                                                                       CONCEN15
      C(N,K,J,I) = C(N,K,J,I) + F1*CP(N,NP)                             CONCEN16
      S(N,K,J,I) = S(N,K,J,I) + F1                                      CONCEN17
      C(N,K,J,II)= C(N,K,J,II)+ F2*CP(N,NP)                             CONCEN18
      S(N,K,J,II)= S(N,K,J,II)+ F2                                      CONCEN19
      C(N,K,JJ,II)=C(N,K,JJ,II)+F3*CP(N,NP)                             CONCEN20
      S(N,K,JJ,II)=S(N,K,JJ,II)+F3                                      CONCEN21
      C(N,K,JJ,I) =C(N,K,JJ,I)+ F4*CP(N,NP)                             CONCEN22
      S(N,K,JJ,I) = S(N,K,JJ,I)+ F4                                     CONCEN23
      C(N,KK,J,I)  = C(N,KK,J,I) + F5*CP(N,NP)                          CONCEN24
      S(N,KK,J,I)  = S(N,KK,J,I) + F5                                   CONCEN25
      C(N,KK,J,II) = C(N,KK,J,II)+ F6*CP(N,NP)                          CONCEN26
      S(N,KK,J,II) = S(N,KK,J,II)+ F6                                   CONCEN27
      C(N,KK,JJ,II)= C(N,KK,JJ,II)+F7*CP(N,NP)                          CONCEN28
      S(N,KK,JJ,II)=S(N,KK,JJ,II)+F7                                    CONCEN29
      C(N,KK,JJ,I) = C(N,KK,JJ,I)+ F8*CP(N,NP)                          CONCEN30
      S(N,KK,JJ,I) = S(N,KK,JJ,I)+ F8                                   CONCEN31
C                                                                       CONCEN32
   50 CONTINUE                                                          CONCEN33
C                                                                       CONCEN34
  100 CONTINUE                                                          CONCEN35
C                                                                       CONCEN36
      DO 200 I=1,NX                                                     CONCEN37
      DO 200 J=1,NY                                                     CONCEN38
      DO 200 K=1,NZ                                                     CONCEN39
      DO 200 N=1,NSP                                                    CONCEN40
      C(N,K,J,I)=C(N,K,J,I)/S(N,K,J,I)                                  CONCEN41
  200 CONTINUE                                                          CONCEN42
C                                                                       CONCEN43
      RETURN                                                            CONCEN44
      END                                                               CONCEN45
```

CONTUR

```
      .CONTUR
      SUBROUTINE CONTUR(IXMAX,IX,IY,X,K,TIXX,KK,TAB,KKK)                    CONTUR01
      COMMON /TFIX/ICYCL,TIME                                               CONTUR02
      DIMENSION TAB(2),XI(IXMAX,IY)                                         CONTUR03
      DATA NC,CMIN,CMAX,LOGFL / 10,0.,0.,0 /                                CONTUR04
C     **ADJUST PRINTER SPACING                                              CONTUR05
      CALL PRTCN(1H ,06,0,0.*)                                              CONTUR06
      CALL PRTCN(1L,4.*)                                                    CONTUR07
      WRITE(6,1000) (TAB(I),I=1,2),ICYCL,TIME                               CONTUR08
      CALL PLINC(A,IA,IY,IXMAX,CMIN,CMAX,LOGFL)                             CONTUR09
C     **RESET NORMAL PRINTER SPACING                                        CONTUR10
      CALL PRTCN(1H ,06,6,6.*)                                              CONTUR11
C                                                                           CONTUR12
      RETURN                                                                CONTUR13
C                                                                           CONTUR14
C     **FORMATS                                                             CONTUR15
C                                                                           CONTUR16
 1000 FORMAT(5X,4H****,3X,2A6,4H****,10X,8HCYCLE = ,I5,6X,7HTIME = ,       CONTUR17
     1   F10.3)                                                             CONTUR18
      END                                                                   CONTUR19
```

DATA

```
      .DATA
      SUBROUTINE DATA                                                    DATA0001
      INCLUDE BLANK                                                      DATA0002
      DIMENSION U1(NX1,NY1),U2(NX1,NY1),U3(NX1,NY1)                      DATA0003
      EQUIVALENCE (U11(1,1),U3(1,1)),(V11(1,1),U2(1,1)),(U11(1,2),       DATA0004
     1   U11(1,1))                                                       DATA0005
      DIMENSION SS(5,NZ1,NY1,NX1), FACTOR(5)                             DATA0006
      EQUIVALENCE (S,SS)                                                 DATA0007
      DATA FACTOR / 30.,96.,0.,0.,13. /                                  DATA0008
      DATA VOLUME /1.036E9/                                              DATA0009
C     **ZERO OUT S ARRAY                                                 DATA0010
      CALL S3ZERO(S,NUM)                                                 DATA0011
C                                                                        DATA0012
      IF(CONST) GO TO 100                                                DATA0013
C*****************TAPE DATA                                            DATA0014
C                                                                        DATA0015
C*****************************                                         DATA0016
   15 READ (10)  DT,U1,U2,U3,EX,EZ                                       DATA0017
      READ (10) (((SS(L,K,J,I),L=1,3),K=1,NZ),J=1,NY),I=1,NX)            DATA0018
C                                                                        DATA0019
C                                                                        DATA0020
      IF(DT .LT. 0.0) STOP NEG DT                                        DATA0021
      RDT=1.0/DT                                                         DATA0022
      DO 20 I=1,NX                                                       DATA0023
      DO 20 J=1,NY                                                       DATA0024
      U(I,J,2)=U(I,J,1)                                                  DATA0025
      U(I,J,3)=-U(I,J,1)                                                 DATA0026
      U(I,J,4)=U(I,J,3)                                                  DATA0027
      V(I,J,2)=V(I,J,1)                                                  DATA0028
      V(I,J,3)=-V(I,J,1)                                                 DATA0029
      V(I,J,4)=V(I,J,3)                                                  DATA0030
      W(I,J,1)=0.0                                                       DATA0031
      W(I,J,2)=2.0*W(I,J,2)                                              DATA0032
      W(I,J,3)=W(I,J,2)                                                  DATA0033
      W(I,J,4)=0.0                                                       DATA0034
      DO 19 K=1,NZ                                                       DATA0035
      EX(I,J,K)=EX(I,J,K)*RDT                                            DATA0036
      EZ(I,J,K)=EZ(I,J,K)*RDT                                            DATA0037
   19 CONTINUE                                                           DATA0038
   20 CONTINUE                                                           DATA0039
C                                                                        DATA0040
C     **MOVE SS ARRAY TO S ARRAY AND CONVERT KG TO PPM                   DATA0041
C                                                                        DATA0042
      DO 50 K=1,NZ                                                       DATA0043
      DO 50 J=1,NY                                                       DATA0044
      DO 50 I=1,NX                                                       DATA0045
      S(5,K,J,I)=SS(3,K,J,I)                                             DATA0046
      S(3,K,J,I)=0.0                                                     DATA0047
C     AIR DENSITY 1.23 KG/M**3; MOLE WEIGHT 28.9                         DATA0048
      DO 45 L=1,NSP                                                      DATA0049
      IF(FACTOR(L) .LT. 1.0E-8) GO TO 45                                 DATA0050
      S(L,K,J,I)=S(L,K,J,I)*28.9*1.0E6*RDT/(FACTOR(L)*1.23*VOLUME)       DATA0051
   45 CONTINUE                                                           DATA0052
   50 CONTINUE                                                           DATA0053
```

```
DATA
      IF(FLAG1 .LT. 1.0E-6) GO TO 200                                   DATA0054
C     **OPTION TO TURN OFF DIFFUSION                                    DATA0055
      CALL S3ZERO(EX,LUM)                                               DATA0056
      CALL S3ZERO(EZ,LUM)                                               DATA0057
      GO TO 200                                                         DATA0058
C                                                                       DATA0059
C********DATA SPECIFIED BY INPUT                                       DATA0060
C                                                                       DATA0061
  100 DO 150 I=1,NX                                                     DATA0062
      DO 150 J=1,NY                                                     DATA0063
      DO 150 K=1,NZ                                                     DATA0064
      EX(I,J,K)=EXFIX                                                   DATA0065
      EZ(I,J,K)=EZFIX                                                   DATA0066
      U(I,J,K)=UFIX                                                     DATA0067
      V(I,J,K)=VFIX                                                     DATA0068
      W(I,J,K)=WFIX                                                     DATA0069
  150 CONTINUE                                                          DATA0070
C                                                                       DATA0071
  200 DTDX=DT*RDX                                                       DATA0072
      DTDY=DT*RDY                                                       DATA0073
      DTDZ=DT*RDZ                                                       DATA0074
C                                                                       DATA0075
      RETURN                                                            DATA0076
      END                                                               DATA0077
```

DEFINE

```
      ,DEFINE
      SUBROUTINE  DEFINE                                                DEFINE01
      INCLUDE BLANK                                                     DEFINE02
      DATA DUMMY/5/                                                     DEFINE03
      NCOMM=74+LUN*(2*NSPI+5)+MAX0(NSPI+3)                              DEFINE04
      NXP1=NX+1                                                         DEFINE05
      NYP1=NY+1                                                         DEFINE06
      NZP1=NZ+1                                                         DEFINE07
      NXM1=NX-1                                                         DEFINE08
      NYM1=NY-1                                                         DEFINE09
      NZM1=NZ-1                                                         DEFINE10
      RDX=1.0/DX                                                        DEFINE11
      RDY=1.0/DY                                                        DEFINE12
      RDZ=1.0/DZ                                                        DEFINE13
      RDX2=RDX*RDX                                                      DEFINE14
      RDY2=RDY*RDY                                                      DEFINE15
      RDZ2=RDZ*RDZ                                                      DEFINE16
      PRINT 100, NCOMM                                                  DEFINE17
  100 FORMAT(1X,'**********NCOMM = ',I6,'**********')                    DEFINE18
      RETURN                                                            DEFINE19
      END                                                               DEFINE20
```

DIFFUS

```
      .DIFFUS
      SUBROUTINE DIFFUS(N)                                                    DIFFUS01
      INCLUDE BLANK                                                           DIFFUS02
C                                                                             DIFFUS03
C                                                                             DIFFUS04
      DO 100 I=1,NX                                                           DIFFUS05
      DO 100 J=1,NY                                                           DIFFUS06
      DO 100 K=1,NZ                                                           DIFFUS07
C                                                                             DIFFUS08
      CALL INDEX(I,J,K)                                                       DIFFUS09
C                                                                             DIFFUS10
C                                                                             DIFFUS11
C     ***** X-Y DIFFUSION                                                     DIFFUS12
C                                                                             DIFFUS13
      TERM1=EX(I,J,K)*((C(N,K,J,IP1)-2.0*C(N,K,J,I)+C(N,K,J,IM1))*RDX2        DIFFUS14
     1   +(C(N,K,JP1,I)-2.0*C(N,K,J,I)+C(N,K,JM1,I))*RDY2)                    DIFFUS15
C                                                                             DIFFUS16
C     ***** Z DIFFUSION                                                       DIFFUS17
C                                                                             DIFFUS18
      TERM2=EZ(I,J,K)*((C(N,KP1,J,I)-2.0*C(N,K,J,I)+C(N,KM1,J,I))*RDZ2        DIFFUS19
C                                                                             DIFFUS20
C     ***** INCREMENT SOURCE ARRAY                                            DIFFUS21
C                                                                             DIFFUS22
      S(N,K,J,I)=S(N,K,J,I)+TERM1+TERM2                                       DIFFUS23
  100 CONTINUE                                                                DIFFUS24
      RETURN                                                                  DIFFUS25
C*************************************************************************** DIFFUS26
      SUBROUTINE INDEX(I,J,K)                                                 DIFFUS27
C                                                                             DIFFUS28
C                                  TEMP B.C.                                  DIFFUS29
      IP1=I+1                                                                 DIFFUS30
      IM1=I-1                                                                 DIFFUS31
      JP1=J+1                                                                 DIFFUS32
      JM1=J-1                                                                 DIFFUS33
      KP1=K+1                                                                 DIFFUS34
      KM1=K-1                                                                 DIFFUS35
      IF(I.EQ.1)       IM1=NX                                                 DIFFUS36
      IF(J.EQ.1)       JM1=NY                                                 DIFFUS37
      IF(K .EQ. 1) KM1=1                                                      DIFFUS38
      IF(I.EQ.NX)      IP1= 1                                                 DIFFUS39
      IF(J.EQ.NY)      JP1= 1                                                 DIFFUS40
      IF(K .EQ. NZ) KP1=NZ                                                    DIFFUS41
C                                                                             DIFFUS42
C                                                                             DIFFUS43
      RETURN                                                                  DIFFUS44
C*************************************************************************** DIFFUS45
      END                                                                     DIFFUS46
```

EDIT

```
      *EDIT
      SUBROUTINE EDIT                                                    EDIT0001
      INCLUDE BLANK                                                      EDIT0002
      DIMENSION CLIN(NX1)                                                EDIT0003
      DIMENSION FACT(5)                                                  EDIT0004
      DATA FACT /4*100.0,1.0/                                            EDIT0005
C                                                                        EDIT0006
C*********************************************************************** *EDIT0007
C                                                                        EDIT0008
C     **ISP SELECTS WHICH SPECIES ARE EDITED                             EDIT0009
C                                                                        EDIT0010
   98 DO 100 I=1,NSP                                                     EDIT0011
      IF(ISP(I) .EQ. 0) GO TO 100                                        EDIT0012
      SP=TITLES(I)                                                       EDIT0013
C                                                                        EDIT0014
      SUM=0.0                                                            EDIT0015
      DO 60 K=1,LEVELS                                                   EDIT0016
      IF(MOD(K+1,2) .EQ. 0) PRINT 6005                                   EDIT0017
 6005 FORMAT(1H1)                                                        EDIT0018
      TIMEX=TIME/3600.                                                   EDIT0019
      PRINT 6010,SP,K,TIMEX,ICYCL                                        EDIT0020
 6010 FORMAT(1X,'CONCENTRATIONS OF ',A6,'  IN X-Y SPACE FOR LEVEL ',I2   EDIT0021
     1   /,5X,'TIME = ',F4.2,4X,'CYCLE = ',I5,/)                         EDIT0022
      DO 50 J = NY, 1, -1                                                EDIT0023
      DO 49 I=1,NX                                                       EDIT0024
      CLIN(I)=FACT(N)*C(N,K,J,I)                                         EDIT0025
      SUM=SUM+CLIN(I)                                                    EDIT0026
   49 CONTINUE                                                           EDIT0027
   50 PRINT 6050, (CLIN(I),I=1,NXM1)                                     EDIT0028
 6050 FORMAT(2X,21F6.2)                                                  EDIT0029
   60 CONTINUE                                                           EDIT0030
      WRITE(6,1000) SUM                                                  EDIT0031
 1000 FORMAT(//,1X,'SUM OF ALL CONCENTRATIONS = ',F12.3)                 EDIT0032
  100 CONTINUE                                                           EDIT0033
C                                                                        EDIT0034
 5000 RETURN                                                             EDIT0035
      END                                                                EDIT0036
```

```
INPUT

      .INPUT
1           SUBROUTINE INPUT                                                  INPUT001
2           INCLUDE BLANK                                                     INPUT002
3     C                                                                       INPUT003
4           NAMELIST /START/RESTRT,ISTART,TMAX,CYCMAX,CHANGE                  INPUT004
5     C                                                                       INPUT005
6     C                                                                       INPUT006
7           DATA NX,NY,NZ,NSP,DX,DY,DZ,CONST / 4,4,1,5,10.,10.,10.,.TRUE. /   INPUT007
8           DATA  NPC,DT,TIME / 1,600.,21600. /                               INPUT008
9           DATA TITLES /6HNO     ,6HNO2    ,6HO3     ,6HHNO2   ,6HHC       /INPUT009
10          DATA ISP /5*1 /                                                   INPUT010
11          NAMELIST /SPECS/ IDUMP,TDUMP,IPRINT,TPRINT,IPLOT,TPLOT,           INPUT011
12         1   LEVELS,I1,I2,I3,I4,I5,NXY,NXZ,NYZ,XYP,XZP,YZP,ISP             INPUT012
13          NAMELIST /GEN/ NX,NY,NZ,NSP,DX,DY,DZ,NPC,NS,TIME,DT,FLAG1,FLAG2, INPUT013
14         1   IFLAG1,IFLAG2,CONST,UFIX,VFIX,WFIX,EXFIX,EZFIX,GROUND          INPUT014
15    C                                                                       INPUT015
16          READ(5,START)                                                     INPUT016
17    C                                                                       INPUT017
18          IF(RESTRT.LT.1.E-6)  GO TO 100                                    INPUT018
19    C                                                                       INPUT019
20    C     TAPE RESTART                                                      INPUT020
21    C                                                                       INPUT021
22          CALL RTAPE                                                        INPUT022
23    C                                                                       INPUT023
24          IF(CHANGE.LT.1.E-6) GO TO 200                                     INPUT024
25      100 CONTINUE                                                          INPUT025
26          READ(5,SPECS)                                                     INPUT026
27          IF(RESTRT.GT.1.E-6) GO TO 200                                     INPUT027
28          READ(5,GEN)                                                       INPUT028
29      200 CONTINUE                                                          INPUT029
30          ICYCNT=CYCMAX                                                     INPUT030
31          RETURN                                                            INPUT031
32          END                                                               INPUT032
```


SSS-R-74-1756

```
      .MAIN
      INCLUDE BLANK,LIST
      COMMON /PANCEL/ II,JJ,KK,F1,F2,F3,F4,F5,F6,F7,F8,I,J,K
C
C
C
      CALL INPUT
C
      IF(RESTRT.GT.0.01) GO TO 10
C
      CALL DEFINE
C     **INITIALIZE ICYCL,SAVE
      ICYCL=0
      SAVE=.F.LSE.
      IF(IDUMP .GT. 0.0  .OR.  IDUMP .GT. 01 SAVE=.TRUE.
C
C     **OBTAIN INITIAL PARTICLE POSITIONS
C
      CALL SETUP
C
C**********************************************************************
C**********************************************************************
   10 ICYCL=ICYCL+1
      TX1=TICKER(1V)
C
C     **SJWARN CHECK
C
      IF(SAVE) CALL SJWARN(20,1,$3001
C
C     **OBTAIN WINDS,DIFFUSIVITIES,AND SOURCES
C
      CALL DATA
C
C     **ADVANCE THE TIME
C
      TIME=TIME+DT
C
C     **PUT DIFFUSION CONTRIBUTIONS IN SOURCE ARRAY
      DO 20 N=1,NSP
      CALL DIFFUS(N)
   20 CONTINUE
C**********************************PARTICLE LOOP*********************INDEX NP
      DO 30 NP=1,NPM
C
C     **OBTAIN VOLUME AVERAGING FACTORS
C
      CALL VOLFAC(NP)
C
      DO 5 N=1,NSP
      DUM(N) = S(N,K,J,I)*F1*S(N,K,J,II)*F2*S(N,K,JJ,II)*F3*
     1         S(N,K,JJ,I)*F4*S(N,KK,J,I)*F5*S(N,KK,J,II)*F6*
     2         S(N,KK,JJ,II)*F7*S(N,KK,JJ,I)*F8
    5 CONTINUE
C
```

```
          CPSUM=0.0                                                          MAIN0054
          DO 8 N=1,NSP                                                       MAIN0055
          CP(N,NP)=CP(N,NP)+DUM(N)*DT                                        MAIN0056
          IF(CP(N,NP) .LT. 0.0) CP(N,NP)=0.0                                 MAIN0057
          CPSUM=CPSUM+CP(N,NP)                                               MAIN0058
        8 CONTINUE                                                           MAIN0059
C                                                                            MAIN0060
C       **PHOTOCHEMICAL CONTRIBUTIONS                                        MAIN0061
C                                                                            MAIN0062
C            AVOID CHEMISTRY IF SUM OF CONCENTRATIONS IS < 0.01 PPM          MAIN0063
          IF(CPSUM .LT. 0.01) GO TO 25                                       MAIN0064
C         PREPARE CP ARRAY FOR CHEMISTRY                                     MAIN0065
          DO 9 N=1,NSP                                                       MAIN0066
          CP(N,NP)=CP(N,NP)+DUM(N)*DT                                        MAIN0067
          IF(CP(N,NP) .LT. 0.0) CP(N,NP)=0.0                                 MAIN0068
        9 CONTINUE                                                           MAIN0069
          CALL CHEM(CP(1,NP),CP(2,NP),CP(3,NP),CP(4,NP),CP(5,NP),DUM(1),     MAIN0070
         1   DUM(2),DUM(3),DUM(4),DUM(5),DT,TIME)                            MAIN0071
C                                                                            MAIN0072
C     **ADVECTION                                                            MAIN0073
C                                                                            MAIN0074
       25 CALL ADVECT                                                        MAIN0075
       30 CONTINUE                                                           MAIN0076
C*********************END PARTICLE LOOP************************** *MAIN0077
C                                                                            MAIN0078
C     **COMPUTE CONCENTRATIONS                                               MAIN0079
C                                                                            MAIN0080
          CALL CONCEN                                                        MAIN0081
C                                                                            MAIN0082
C     **DETERMINE IF THIS CYCLE'S RESULTS ARE TO BE PLOTTED,                 MAIN0083
C          PRINTED, OR DUMPED                                              MAIN0084
C                                                                            MAIN0085
          PLOT=.FALSE.                                                       MAIN0086
          PRINT=.FALSE.                                                      MAIN0087
          DUMP=.FALSE.                                                       MAIN0088
          CALL PRTTST                                                        MAIN0089
C     **SELECT OUTPUT ROUTINES                                               MAIN0090
          IF(PRINT) CALL EDIT                                                MAIN0091
          IF(PLOT) CALL OUTPUT                                               MAIN0092
          IF(DUMP) CALL ATAPE                                                MAIN0093
          IF(.NOT. GROUND) GO TO 100                                         MAIN0094
C     **WRITE OUT GROUND LEVEL CONCENTRATIONS                                MAIN0095
          WRITE (11)  ICYCL,TIME                                             MAIN0096
          DO 75 N=1,NSP                                                      MAIN0097
          WRITE (11)  ((CON(N,I,J,1),J=1,NY),I=1,NX)                         MAIN0098
       75 CONTINUE                                                           MAIN0099
          PRINT 1250, ICYCL                                                  MAIN0100
          TX2=TICKER(TV)                                                     MAIN0101
          TELAPS=TX2-TX1                                                     MAIN0102
          PRINT 1200,TIME,DT,TELAPS                                          MAIN0103
 1200 FORMAT(1X,'TIME = ',F10.3,4X,'DT = ',F8.3,4X,'ELAPSED = ',F9.5)        MAIN0104
 1250 FORMAT(1X,'CYCLE = ',I4,5X,'***LEVEL 1 CONCENTRATIONS DUMPED****      MAIN0105
C                                                                            MAIN0106
C     **IS THIS RUN COMPLETED?                                               MAIN0107
```

```
C
  100 IF(TIME .GE. TMAX .AND. TMAX .GT. 0.0) GO TO 200                  MAIN108
      IF(ICYCL .GE. CYCMAX .AND. CYCMAX .GT. 0.0) GO TO 200             MAIN109
C                                                                       MAIN110
C     **RETURN TO MAIN LOOP FOR ANOTHER CYCLE                           MAIN111
C                                                                       MAIN112
      GO TO 10                                                          MAIN113
C*****************************************************************    MAIN114
C*****************************************************************    *MAIN115
C*****************************************************************    *MAIN116
C                                                                       MAIN117
C     **NORMAL EXIT                                                     MAIN118
  200 IF(SAVE .AND. (.NOT. PUMP)) CALL WTAPE                            MAIN119
      IF(.NOT. PLOT) CALL OUTPUT                                        MAIN120
      IF(.NOT. PRINT) CALL EDIT                                         MAIN121
      STOP NORMAL                                                       MAIN122
C     **SSWAP: TERMINATION                                              MAIN123
  300 CALL WTAPE                                                        MAIN124
      CALL EDIT                                                         MAIN125
      STOP SSWARN                                                       MAIN126
C                                                                       MAIN127
C                                                                       MAIN128
      END                                                               MAIN129
```

OUTPUT

```
      .OUTPUT
      SUBROUTINE OUTPUT                                                  OUTPUT01
      INCLUDE BLANK                                                      OUTPUT02
      COMMON /TFIX/ IGG,TGG                                              OUTPUT03
      DIMENSION BIGPLT(200),XYPL(NX1,NY1),XZPL(NX1,NZ1),YZPL(NY1,NZ1)    OUTPUT04
      DIMENSION TAB(2)                                                   OUTPUT05
      EQUIVALENCE  (XYPL(1,1),BIGPLT(1)),(XZPL(1,1),BIGPLT(1)),          OUTPUT06
     1   (YZPL(1,1),BIGPLT(1))                                           OUTPUT07
C                                                                        OUTPUT08
C                                                                        OUTPUT09
      IGG=ICYCL                                                          OUTPUT10
      TGG=TIME                                                           OUTPUT11
C                                                                        OUTPUT12
C     **DETERMINE CONTOUR PLOTS DESIRED                                  OUTPUT13
C                                                                        OUTPUT14
      DO 1000 N=1,NSP                                                    OUTPUT15
      NQ=N                                                               OUTPUT16
C                                                                        OUTPUT17
C     **ISP SELECTS WHICH SPECIES ARE PLOTTED                            OUTPUT18
      TAB(1)=TITLES(N)                                                   OUTPUT19
      TAB(2)=GMPLOT                                                      OUTPUT20
      IF(ISP(N) .EQ. 0) GO TO 1000                                       OUTPUT21
      IF(NXY .EQ. 0) GO TO 100                                           OUTPUT22
C**********X-Y CONTOUR PLOTS                                            OUTPUT23
      DO 50 L=1,NXY                                                      OUTPUT24
C                                                                        OUTPUT25
      DO 10 I=1,NX                                                       OUTPUT26
      DO 10 J=1,NY                                                       OUTPUT27
      LEVEL=XYP(L)                                                       OUTPUT28
      XYPL(I,J)=C(NQ,LEVEL,J,I)                                          OUTPUT29
   10 CONTINUE                                                           OUTPUT30
      CALL CONTUR(NX1,NX,NY,XYPL,0,T,0,TAB,1)                            OUTPUT31
   50 CONTINUE                                                           OUTPUT32
  100 IF(NXZ .EQ. 0) GO TO 200                                           OUTPUT33
C**********X-Z CONTOUR PLOTS                                            OUTPUT34
      DO 150 L=1,NXZ                                                     OUTPUT35
C                                                                        OUTPUT36
      DO 110 I=1,NX                                                      OUTPUT37
      DO 110 K=1,NZ                                                      OUTPUT38
      LEVEL=XZP(L)                                                       OUTPUT39
      XZPL(I,K)=C(NQ,K,LEVEL,I)                                          OUTPUT40
  110 CONTINUE                                                           OUTPUT41
      CALL CONTUR(NX1,NX,NZ,XZPL,0,T,0,TAB,1)                            OUTPUT42
  150 CONTINUE                                                           OUTPUT43
  200 IF(NYZ .EQ. 0) GO TO 300                                           OUTPUT44
C**********Y-Z CONTOUR PLOTS                                            OUTPUT45
      DO 250 L=1,NYZ                                                     OUTPUT46
C                                                                        OUTPUT47
      DO 210 J=1,NY                                                      OUTPUT48
      DO 210 K=1,NZ                                                      OUTPUT49
      LEVEL=YZP(L)                                                       OUTPUT50
      YZPL(J,K)=C(NQ,K,J,LEVEL)                                          OUTPUT51
  210 CONTINUE                                                           OUTPUT52
      CALL CONTUR(NY1,NY,NZ,YZPL,0,T,0,TAB,1)                            OUTPUT53
```

```
      OUTPUT
      OUTPUT54
      OUTPUT55
      OUTPUT56
      OUTPUT57
      OUTPUT58
      OUTPUT59
      OUTPUT60
      OUTPUT61
      OUTPUT62
C
C     250 CONTINUE
C
C     300 CONTINUE
C
C     1000 CONTINUE
C
      RETURN
C
      END
```

PL

```
      .PL
      SUBROUTINE PL(NNC, Z, NX, NY, NXMX, AMIN, AMAX, LOGFL)                    PL000001
      DIMENSION LV(72), L(131), Z(NXMX,NY)                                      PL000002
      DATA LQ / ' ', '1', '2', '3', '4', '5',                                   PL000003
     1 '6', '7', '8', '9', 'A', 'B',                                            PL000004
     2 'C', 'D', 'E', 'F', 'G', 'H', 'I',                                       PL000005
     3 'J', 'K', 'L', 'M', 'N', 'O', 'P',                                       PL000006
     4 'Q', 'R', 'S', 'T', 'U', 'V',                                            PL000007
     5 'W', 'X', 'Y', 'Z', ' ' /                                                PL000008
      DATA ISTAR / '*' /, IMAX / 131 /                                          PL000009
      JMAX = .6*NY/NX*IMAX + .5                                                 PL000010
      NC = MIN(NNC, 36)                                                         PL000011
      ZMIN = AMIN                                                               PL000012
      ZMAX = AMAX                                                               PL000013
      ANX = NX                                                                  PL000014
      ANY = NY                                                                  PL000015
      IF(ABS(ZMAX-ZMIN) .GT. 1.0E-5) GO TO 20                                   PL000016
      ZMAX = Z(1,1)                                                             PL000017
      ZMIN = ZMAX                                                               PL000018
      DO 10 I = 1, NX                                                           PL000019
      DO 10 J = 1, NY                                                           PL000020
      ZMAX = AMAX1(ZMAX, Z(I,J))                                                PL000021
   10 ZMIN = AMIN1(ZMIN, Z(I,J))                                                PL000022
      IF(ABS(ZMAX-ZMIN) .LT. 1.0E-20) RETURN                                    PL000023
   20 IF (LOGFL .EQ. 0) GO TO 30                                                PL000024
      IF (ZMIN .GT. 0.) GO TO 25                                                PL000025
      PRINT 6020                                                                PL000026
 6020 FORMAT( ///// ' **********NON-POSITIVE VALUE IN ARRAY  0PL000027
     1 RE CONTOURED LOGARITHMICALLY.  PL IS RETURNING WITHOUT PLOTTING 1PL000028
      RETURN                                                                    PL000029
   25 HZMAX = ZMAX                                                              PL000030
      HZMIN = ZMIN                                                              PL000031
      ZMAX = ALOG10(ZMAX)                                                       PL000032
      ZMIN = ALOG10(ZMIN)                                                       PL000033
   30 SC = (NC*2. - 1.E-5)/(ZMAX - ZMIN)                                        PL000034
      DX = ANX/IMAX                                                             PL000035
      DY = ANY/JMAX                                                             PL000036
      Y = ANY + DY                                                              PL000037
      DO 80 J = JMAX, 1, -1                                                     PL000038
      Y = Y - DY                                                                PL000039
      M = Y                                                                     PL000040
      IF (M .LE. 0) M = 1                                                       PL000041
      IF (M .GE. NY) M = NY - 1                                                 PL000042
      DM = Y - M                                                                PL000043
      X = 0.                                                                    PL000044
      DO 70 I = 1, IMAX                                                         PL000045
      X = X + DX                                                                PL000046
      IF (I*J .EQ. 1 .OR. I*J .EQ. IMAX*JMAX) GO TO 60                          PL000047
      IF (I .EQ. 1 .AND. J .EQ. JMAX .OR. J .EQ. 1 .AND.                        PL000048
     1 I .EQ. IMAX) GO TO 60                                                    PL000049
      N = X                                                                     PL000050
      IF (N .LE. 0) N = 1                                                       PL000051
      IF (N .GE. NX) N = NX - 1                                                 PL000052
      DN = X - N                                                                PL000053
```

```
PL

      IF (LOGFL .NE. 0) GO TO 40                                          PL000059
      C = DN*(DM*Z(N+1,M+1) + (1. - DM)*Z(N+1,M)) + (1. - DN)*            PL000055
     1 (DM*Z(N,M+1) + (1. - DM)*Z(N,M))                                   PL000056
      GO TO 50                                                            PL000057
   40 C = DN*(DM*ALOG10(Z(N+1,M+1)) + (1. - DM)*ALOG10(Z(N+1,M))) +       PL000058
     1 (1. - DN)*(DM*ALOG10(Z(N,M+1)) + (1. - DM)*ALOG10(Z(N,M)))         PL000059
   50 IND = 1 + SC*(C - ZMIN)                                             PL000060
      IF (IND .LE. 0) IND = 1                                             PL000061
      IF(IND .GT. ZONC) IND=ZONC                                          PL000062
      L(I) = LQ(IND)                                                      PL000063
      GO TO 70                                                            PL000064
   60 L(I) = ISTAR                                                        PL000065
   70 CONTINUE                                                            PL000066
   80 PRINT 6080, (L(I),I=1,IMAX)                                         PL000067
 6080 FORMAT (1X, 131A1)                                                  PL000068
      IF (LOGFL .EQ. 0) GO TO 90                                          PL000069
      ZMAX = HZMAX                                                        PL000070
      ZMIN = HZMIN                                                        PL000071
   90 PRINT 6090, ZMIN, ZMAX                                              PL000072
 6090 FORMAT (*0MINIMUM VALUE = *, 1PE9.3, 5X, *MAXIMUM VALUE = *, E9. 1PLC00073
C                                                                         PLC00074
      RETURN                                                              PL000075
      END                                                                 PLC00076
```

```
          .PRTTST
1               SUBROUTINE PRTIST                                                  PRTTST01
2               INCLUDE BLANK                                                      PRTTST02
3                                                                                  PRTTST03
4         C                                                                        PRTTST04
5         C     **THIS SUBROUTINE CONTROLS PRINTER EDITS, PLOTS, AND TAPE DUMPS    PRTTST05
6         C                                                                        PRTTST06
7         C     *****PLOTS                                                         PRTTST07
8               IF(IPLOT .EQ. 0) GO TO 30                                          PRTTST08
9               IF(MOD(ICYCL,IPLOT) .GT. 0.1) GO TO 30                             PRTTST09
10              PLOT=.TRUE.                                                        PRTTST10
11            30 IF(TPLOT .LT. 1.0E-8) GO TO 50                                    PRTTST11
12              DUM(5)=TIMO+TPLOT                                                  PRTTST12
13              IF(TIME .LT. DUM(5)) GO TO 50                                      PRTTST13
14              TIMO=DUM(5)                                                        PRTTST14
15              PLOT=.TRUE.                                                        PRTTST15
16        C     *****PRINTER EDITS                                                 PRTTST16
17            50 IF(IPRINT .EQ. 0) GO TO 60                                        PRTTST17
18              IF(MOD(ICYCL,IPRINT) .GT. 0.1) GO TO 60                            PRTTST18
19              PRINTX=.TRUE.                                                      PRTTST19
20            60 IF(TPRINT .LT. 1.0E-8) GO TO 80                                   PRTTST20
21              DUM(6)=TIMO+TPRINT                                                 PRTTST21
22              IF(TIME .LT. DUM(6)) GO TO 80                                      PRTTST22
23              TIMO=DUM(6)                                                        PRTTST23
24              PRINTX=.TRUE.                                                      PRTTST24
25        C     *****TAPE DUMPS                                                    PRTTST25
26            80 IF(IDUMP .EQ. 0) GO TO 90                                         PRTTST26
27              IF(MOD(ICYCL,IDUMP) .GT. 0.1) GO TO 90                             PRTTST27
28              DUMP=.TRUE.                                                        PRTTST28
29            90 IF(TDUMP .LT. 1.0E-8) GO TO 100                                   PRTTST29
30              DUM(7)=TIMO+TDUMP                                                  PRTTST30
31              IF(TIME .LT. DUM(7)) GO TO 100                                     PRTTST31
32              TIMO=DUM(7)                                                        PRTTST32
33              DUMP=.TRUE.                                                        PRTTST33
34        C     **ALWAYS PLOT CYCLE ONE, AND DUMP IF DUMPS HAVE BEEN REQUESTED     PRTTST34
35           100 IF(ICYCL .NE. 1) GO TO 150                                        PRTTST35
36              PLOT=.TRUE.                                                        PRTTST36
37              PRINTX=.TRUE.                                                      PRTTST37
38              IF(SAVE) DUMP=.TRUE.                                               PRTTST38
39        C                                                                        PRTTST39
40           150 RETURN                                                          PRTTST40
41               END                                                             PRTTST41
```

```
      .RTAPE
      SUBROUTINE  RTAPE                                                 RTAPE001
      INCLUDE BLANK                                                     RTAPE002
C                                                                       RTAPE003
C     **SAVE RESTARTING DATA                                            RTAPE004
      SAVE1=RESTRT                                                      RTAPE005
      SAVE2=TMAX                                                        RTAPE006
      SAVE3=CYCMAX                                                      RTAPE007
      ISAVE1=ISTART                                                     RTAPE008
C                                                                       RTAPE009
      WRITE(6,1000) ISTART                                              RTAPE010
   25 READ (12)  ICYCL,TIME,NCOMM                                       RTAPE011
      WRITE(6,1016) ICYCL,TIME                                          RTAPE012
      IF(ICYCL .EQ. ISTART) GO TO 100                                   RTAPE013
      READ (12)                                                         RTAPE014
      GO TO 25                                                          RTAPE015
C                                                                       RTAPE016
C     **READ BLANK COMMON                                               RTAPE017
C                                                                       RTAPE018
  100 READ(12) (DUM(I), I=1,NCOMM)                                      RTAPE019
C                                                                       RTAPE020
C     **RESTORE RESTARTING DATA                                         RTAPE021
      RESTRT=SAVE1                                                      RTAPE022
      TMAX=SAVE2                                                        RTAPE023
      CYCMAX=SAVE3                                                      RTAPE024
      ISTART=ISAVE1                                                     RTAPE025
C                                                                       RTAPE026
C     **GET SETUP TAPE SPACED CORRECTLY FOR RESTART                     RTAPE027
C                                                                       RTAPE028
      TTEST=21600.                                                      RTAPE029
C          SPACE PAST INITIAL CONCENTRATIONS                            RTAPE030
      READ (10) DUMM                                                    RTAPE031
C                                                                       RTAPE032
  125 READ (10) DTT,DUMM,DUMM,DUMM,DUMM,DUMM                            RTAPE033
      READ (10) DUMM                                                    RTAPE034
      TTEST=TTEST+DTT                                                   RTAPE035
      CHECK=ABS(TTEST-TIME)                                             RTAPE036
      IF(CHECK .LT. 1.0E-8) GO TO 140                                   RTAPE037
      GO TO 125                                                         RTAPE038
  140 PRINT 1700,TTEST                                                  RTAPE039
C                                                                       RTAPE040
C                                                                       RTAPE041
C                                                                       RTAPE042
C     **GET GROUND LEVEL CONCENTRATION FILE SPACED CORRECTLY            RTAPE043
C                                                                       RTAPE044
      IF(.NOT. GROUND) GO TO 200                                        RTAPE045
  150 READ (11) ITEST,TTEST                                             RTAPE046
      DO 160 N=1,NSP                                                    RTAPE047
      READ (11)                                                         RTAPE048
  160 CONTINUE                                                          RTAPE049
      IF(ITEST .EQ. ISTART) GO TO 190                                   RTAPE050
      GO TO 150                                                         RTAPE051
  190 PRINT 1600                                                        RTAPE052
C                                                                       RTAPE053
```

```
      200 RETURN                                                        RTAPE054
C                                                                       RTAPE055
C     **FORMATS                                                         RTAPE056
C                                                                       RTAPE057
 1000 FORMAT(1H1,///,93H **********TAPE RESTART REQUESTED FOR CYCLE ,I5 RTAPE058
     1   2X,10H********** ,///)                                         RTAPE059
 1010 FORMAT(1X, 6HCYCLE ,I5,2X,19HFOUND. TIME = ,1PE14.6)              RTAPE060
 1700 FORMAT(1X,'SETUP TAPE SPACED FOR RESTART. TTEST = ',F10.3)        RTAPE061
 1600 FORMAT(1X,'GROUND LEVEL CONC FILE SPACED FOR RESTART')            RTAPE062
      END                                                               RTAPE063
```

```
      .SETUP
      SUBROUTINE SETUP                                                  SETUP001
      INCLUDE  BLANK                                                    SETUP002
      COMMON /PARCEL/ II,JJ,KK,F1,F2,F3,F4,F5,F6,F7,F8,I,J,K           SETUP003
      DIMENSION CC(5,NZ),NY),NX))                                       SETUP004
      EQUIVALENCE (C,CC)                                                SETUP005
C     **INITIALIZE CONCENTRATIONS                                       SETUP006
      CALL SSZERO(C,NUM)                                                SETUP007
C                                                                       SETUP008
C                                                                       SETUP009
    9 NP=0                                                              SETUP010
C                                                                       SETUP011
      READ (10) (((CC(L,K,J,I),L=1,3),K=1,NZ),J=1,NY),I=1,NX)          SETUP012
C                                                                       SETUP013
      DO 10 K=1,NZ                                                      SETUP014
      DO 10 J=1,NY                                                      SETUP015
      DO 10 I=1,NX                                                      SETUP016
      C(5,K,J,I)=CC(3,K,J,I)                                            SETUP017
      C(3,K,J,I)=0.0                                                    SETUP018
   10 CONTINUE                                                          SETUP019
      WRITE (11) ICYCL,TIME                                             SETUP020
      DO 22 N=1,NSP                                                     SETUP021
      WRITE (11)  ((C(N,I,J,I),J=1,NY),I=1,NX)                          SETUP022
   22 CONTINUE                                                          SETUP023
C                                                                       SETUP024
      PRINT 1250, ICYCL                                                 SETUP025
 1250 FORMAT(1X,'CYCLE = ',I9,9X,'**LEVEL I CONCENTRATIONS DUMPED**')   SETUP026
C                                                                       SETUP027
C                                                                       SETUP028
C     LOOP OVER CELLS                                                   SETUP029
C                                                                       SETUP030
      DO 50 KP=1,NZ                                                     SETUP031
      DO 50 IP=1,NX                                                     SETUP032
      DO 50 JP=1,NY                                                     SETUP033
      NPCC=NPC                                                          SETUP034
      DO 40 L=1,NPCC                                                    SETUP035
      IF(L .EQ. 2  .AND.  KP .GT. 2) GO TO 40                           SETUP036
      NP=NP+1                                                           SETUP037
      X(NP)=IP                                                          SETUP038
      Y(NP)=JP                                                          SETUP039
      Z(NP)=KP                                                          SETUP040
C     **COMPUTE VOLUME FACTORS                                          SETUP041
      CALL VOLFAC(NP)                                                   SETUP042
C     **COMPUTE INITIAL PARCEL CONCENTRATIONS                           SETUP043
      DO 30 NN=1,NSP                                                    SETUP044
C                                                                       SETUP045
      CP(NN,NP)= C(NN,K,J,I)*F1+C(NN,K,J,II)*F2+C(NN,K,JJ,II)*F3       SETUP046
     1          + C(NN,K,JJ,I)*F4+C(NN,KK,J,I)*F5+C(NN,KK,J,II)*F6     SETUP047
     2          + C(NN,KK,JJ,II)*F7+C(NN,KK,JJ,I)*F8                   SETUP048
   30 CONTINUE                                                          SETUP049
   40 CONTINUE                                                          SETUP050
      IF (NP.GT.MAX) RETURN 0                                           SETUP051
   50 CONTINUE                                                          SETUP052
      NPM=NP                                                            SETUP053
```

```
SETUP
      PRINT 151, NPM                                                    SETUP054
  151 FORMAT(1X,'*****NUMBER OF PARTICLES =',I6)                        SETUP055
C                                                                       SETUP056
      CALL EDIT                                                         SETUP057
C                                                                       SETUP058
      RETURN                                                            SETUP059
C                                                                       SETUP060
C     **FORMATS                                                         SETUP061
C                                                                       SETUP062
C                                                                       SETUP063
C     **FORMATS                                                         SETUP064
C                                                                       SETUP065
 1000 FORMAT()                                                          SETUP066
 1100 FORMAT(/,1X,'USER: REPLY YES TO ENTER INITIAL CONCENTRATIONS  ')  SETUP067
 1110 FORMAT(1X,'USER: ENTER I,J,K  ')                                  SETUP068
 1120 FORMAT(1X,'USER: ENTER C(',3I3,') VALUES  ')                      SETUP069
 1105 FORMAT(A6)                                                        SETUP070
      END                                                               SETUP071
```

VOLFAC

```
      *VOLFAC
      SUBROUTINE *OLFAC(N)                                                  VOLFAC01
      INCLUDE BLANK                                                         VOLFAC02
      COMMON /PARCEL/ II,JJ,KK,F1,F2,F3,F4,F5,F6,F7,F8,I,J,K                VOLFAC03
C                                                                           VOLFAC04
      I=X(N)                                                                VOLFAC05
      J=Y(N)                                                                VOLFAC06
      K=Z(N)                                                                VOLFAC07
      II=I+1                                                                VOLFAC08
      JJ=J+1                                                                VOLFAC09
      KK=K+1                                                                VOLFAC10
      FX1=X(N)-I                                                            VOLFAC11
      FY1=Y(N)-J                                                            VOLFAC12
      FZ1=Z(N)-K                                                            VOLFAC13
      FX=1.-FX1                                                             VOLFAC14
      FY=1.-FY1                                                             VOLFAC15
      FZ=1.-FZ1                                                             VOLFAC16
      F1=FX*FY*FZ                                                           VOLFAC17
      F2=FX1*FY*FZ                                                          VOLFAC18
      F3=FX1*FY1*FZ                                                         VOLFAC19
      F4=FX*FY1*FZ                                                          VOLFAC20
      F5=FX*FY*FZ1                                                          VOLFAC21
      F6=FX1*FY*FZ1                                                         VOLFAC22
      F7=FX1*FY1*FZ1                                                        VOLFAC23
      F8=FX*FY1*FZ1                                                         VOLFAC24
      IF(II.EQ.NAP1) II=1                                                   VOLFAC25
      IF(JJ.EQ.NYP1) JJ=1                                                   VOLFAC26
      IF(I .EQ. 0) I=NX                                                     VOLFAC27
      IF(J .EQ. 0) J=NY                                                     VOLFAC28
      IF(K .NE. 0) GO TO 50                                                 VOLFAC29
      F5=F5+F1                                                              VOLFAC30
      F6=F6+F2                                                              VOLFAC31
      F7=F7+F3                                                              VOLFAC32
      F8=F8+F4                                                              VOLFAC33
      F1=0.0                                                                VOLFAC34
      F2=0.0                                                                VOLFAC35
      F3=0.0                                                                VOLFAC36
      F4=0.0                                                                VOLFAC37
      GO TO 100                                                             VOLFAC38
   50 IF(KK .NE. NZP1) GO TO 100                                            VOLFAC39
      F1=F1+F5                                                              VOLFAC40
      F2=F2+F6                                                              VOLFAC41
      F3=F3+F7                                                              VOLFAC42
      F4=F4+F8                                                              VOLFAC43
      F5=0.0                                                                VOLFAC44
      F6=0.0                                                                VOLFAC45
      F7=0.0                                                                VOLFAC46
      F8=0.0                                                                VOLFAC47
C                                                                           VOLFAC48
  100 RETURN                                                                VOLFAC49
      END                                                                   VOLFAC50
```

```
      *STAPE                                                            *TAPE0001
            SUBROUTINE *TAPE                                            *TAPE0002
            INCLUDE BLANK                                               *TAPE0003
      C                                                                 *TAPE0004
            MTAPE=12                                                    *TAPE0005
            WRITE(MTAPE) ICYCL,TIME,NCOMM                               *TAPE0006
            WRITE(MTAPE) (DUM(I),I=1,NCOMM)                             *TAPE0007
            WRITE(6,1000) ICYCL,TIME                                    *TAPE0008
      C                                                                 *TAPE0009
            RETURN                                                      *TAPE0010
      C                                                                 *TAPE0011
      1000 FORMAT(1X,      '******CYCLE = ',I5,2X,'TIME = ',1PE14.6,2X, *TAPE0012
           1     'DUMPED ON TAPE.******')                               *TAPE0013
            END
```

```
*TAPE                                                          *TAPE001
      SUBROUTINE *TAPE                                         *TAPE002
      INCLUDE BLANK                                            *TAPE003
C                                                              *TAPE004
      MTAPE=12                                                 *TAPE005
      WRITE(MTAPE) ICYCL,TIME,NCOMM                            *TAPE006
      WRITE(MTAPE) (DUM(I),I=1,NCOMM)                          *TAPE007
      WRITE(6,1000) ICYCL,TIME                                 *TAPE008
C                                                              *TAPE009
      RETURN                                                   *TAPE010
C                                                              *TAPE011
 1000 FORMAT(1X,     '*****CYCLE = ',I5,2X,'TIME = ',1PE19.6,2X,*TAPE012
     1     'DUMPED ON TAPE.*****')                             *TAPE013
      END
```

4. A SAMPLE TEST CALCULATION

In this section a test calculation is presented, including copies of the computer print output.

The calculation considered is the advection of a Gaussian distribution, previously considered in Reference 1, page 32.

4.1 CODE CHANGES

Some modifications to the NEXUS/L code used in the Los Angeles photochemistry simulation and documented in the report were made. In particular,

(a) the structure of blank common was reorganized to permit a grid 50 x-cells by 30 y-cells. Thus, the procedure BLANK was modified so that NXI=50, NYI=30, NZI=1, NSPI=1, MAX=9000.

(b) a change to subroutine SETUP was made to properly initialize the concentrations to the desired distribution,

$$c(x,y) = 71.1 \exp\{-[(x-9)^2 + (y-9)^2]/12.5\}$$

Here, $\sigma_x = \sigma_y = 2.5$ km. Also, as before, the concentration at the center of cell <9,9> is set to 69.0.

(c) subroutine SETUP was further changed to permit the initial positions of some of the Lagrangian particles to be away from the cell center. For the fair Lagrangian parcels per cell case documented here, the initial positions for cell I,J are

(i) $x = I - 0.3$
$y = J + 0.3$

(ii) $x = I + 0.3$
$y = J - 0.3$

(iii) $x = I - 0.3$
$y = J - 0.3$

(iv) $x = I - 0.3$
$y = J + 0.3$

Figure 4 displays the Fortran coding of subroutine SETUP used.

4.2 INPUT TO GENERATE TEST CALCULATION

The problem configuration was maintained exactly as the previous Gaussian advection reported in Reference 1.

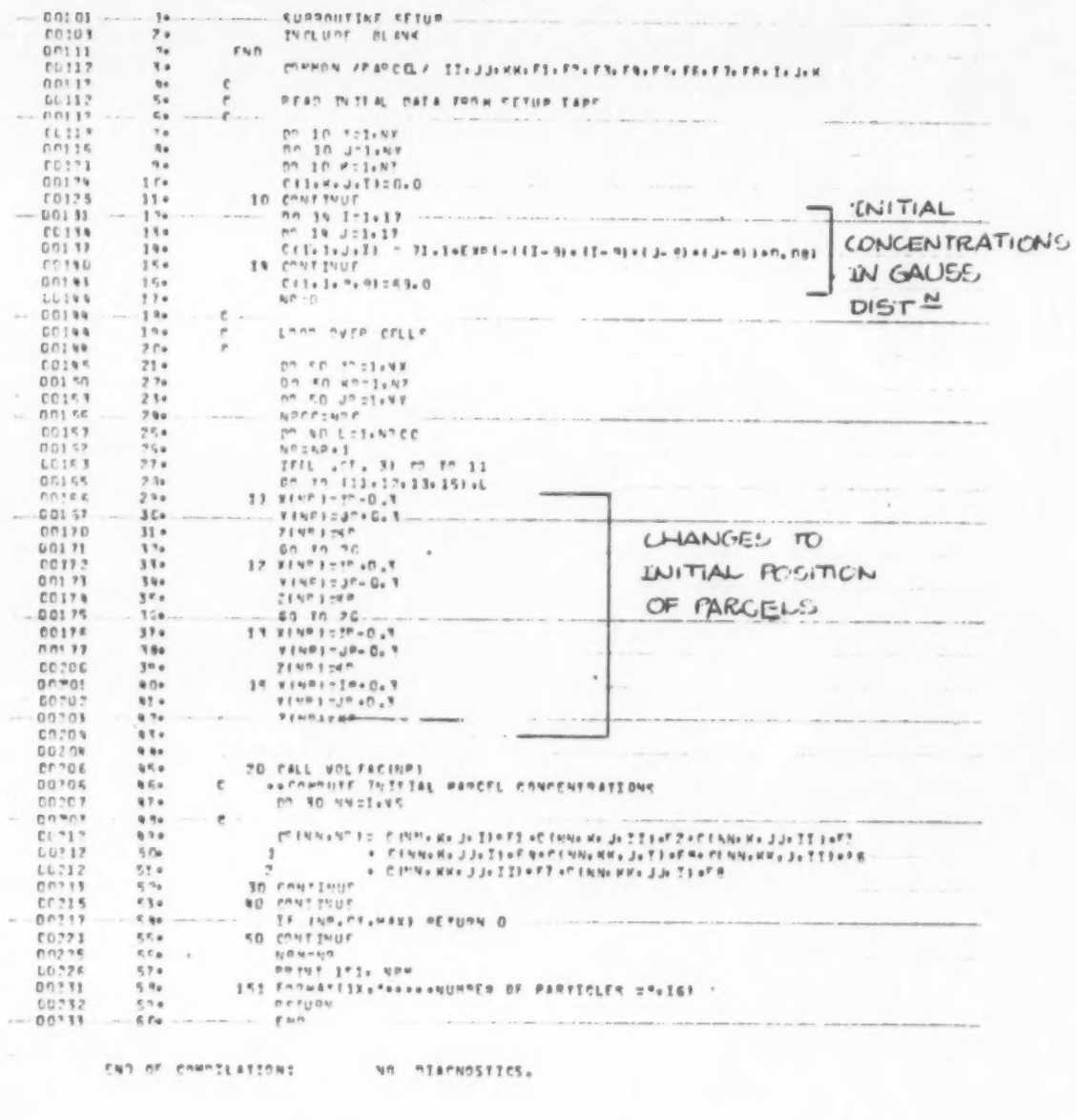
(1) The cells are 1 km cubes, (DX=DY=DZ=1000 m).

(2) There are 50 x-direction cells, 30 y-direction cells. There is only one level (NX=50, NY=30, NZ=1).

(3) The u-velocity is everywhere 10 m/sec, the v-velocity is everywhere 5 m/sec, creating an off-axis net wind direction. (UFIX=10, VFIX=5, WFIX=0)

(4) The initial Gaussian distribution is set to zero in all cells more than eight cells from the center of the distribution.

Reproduced from best available copy.




```
      SUBROUTINE SETUP
      INCLUDE BLANK
      END
      COMMON /PARCEL/ II,JJ,KK,F1,F2,F3,F8,F9,F6,F7,F8,I,J,K
C
C     READ INITIAL DATA FROM SETUP TAPE
C
      DO 10 I=1,NX
      DO 10 J=1,NY
      DO 10 K=1,NZ
      C(I,K,J,1)=0.0
   10 CONTINUE
      DO 15 I=1,17
      DO 15 J=1,17
      C(I,1,J,1) = 71.1*EXP(-((I-9)*(I-9)+(J-9)*(J-9))*0.08)
   15 CONTINUE
      C(1,1,9,9)=69.0
      NP=0
C
C     LOOP OVER CELLS
C
      DO 50 I=1,NX
      DO 50 K=1,NZ
      DO 50 J=1,NY
      NPCC=NPC
      DO 40 L=1,NPCC
      NP=NP+1
      IF(L .GT. 31) GO TO 11
      GO TO (12,13,13,15),L
   12 XINP(L)=IP-0.5
      YINP(L)=JP-0.5
      ZINP(L)=KP
      GO TO 20
   13 XINP(L)=IP+0.5
      YINP(L)=JP-0.5
      ZINP(L)=KP
      GO TO 20
   15 XINP(L)=IP-0.5
      YINP(L)=JP+0.5
      ZINP(L)=KP
C
C
   20 CALL VOLFRC(NP)
C     **COMPUTE INITIAL PARCEL CONCENTRATIONS
      DO 30 NN=1,NS
C
      C(I,NN,J,1)= C(IP,K,JP,I)*F1 +C(IN,K,JP,II)*F2+C(IN,K,JJ,II)*F3
     1           + C(IN,K,JJ,I)*F8+C(IN,KK,JP,I)*F9+C(IN,KK,JP,II)*F6
     2           + C(IN,KK,JJ,II)*F7 +C(IN,KK,JJ,I)*F8
   30 CONTINUE
   40 CONTINUE
      IF (NP.GT.MAX) RETURN 0
   50 CONTINUE
      NPP=NP
      PRINT 151, NPP
  151 FORMAT(1X,'*****NUMBER OF PARTICLES =',I6)
      RETURN
      END

END OF COMPILATION:       NO DIAGNOSTICS.
```

Figure 4. Subroutine SETUP version used for Gaussian advection test problem.

(5) The time step is 400 sec/cycle (DT=400).

(6) There is no diffusion, only advection (EXFIX=0, EZFIX=0).

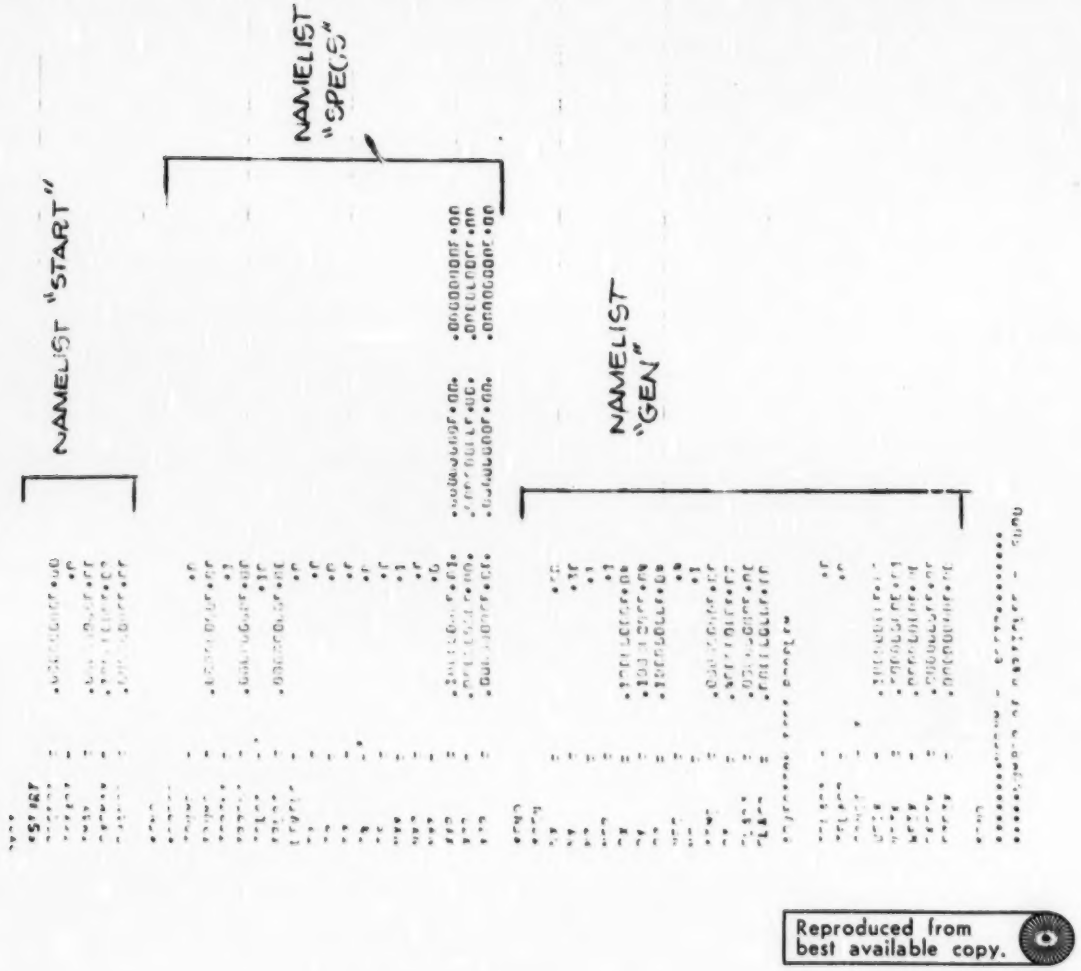
The namelist input used to generate this test calculation is shown in Figure 5.

4.3 CYCLE 1 OUTPUT

Figure 6 displays the printer putput, cell-by-cell, of the problem configuration resulting at the end of Cycle 1. Figure 7 presents a contour plot representation of the same data.

REFERENCE

Sklarew, R.C., et.al., "A Particle-In-Cell Method for Numerical Solution of the Atmospheric Diffusion Equation, And Application to Air Pollution Problems," Report No. 3SR-844 (November, 1971), Systems, Science and Software, La Jolla, Calif.



Reproduced from best available copy.

Figure 5. Namelist input to generate test calculation.

Reproduced from
best available copy.

Reproduced from
best available copy.

Figure 7. Printer contour representation of concentrations at end of cycle 1.